BODY TEMPERATURE DETECTION SYSTEM FOR MALARIA FEVER PATIENT USING RFID AND GSM NETWORK

By

MOHAMAD SYUKRI EFFENDY B. MOHD. YUSOF

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

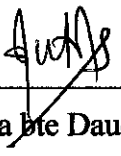
BODY TEMPERATURE DETECTION SYSTEM FOR MALARIA FEVER PATIENT USING RFID AND GSM NETWORK

by

Mohamad Syukri Effendy B. Mohd. Yusof

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



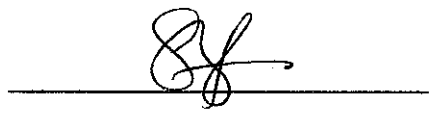
Puan Hanita bte Daud
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



Mohamad Syukri Effendy. Mohd. Yusof

ABSTRACT

Malaria fever patients are known to suffer from fever spikes, where the body temperature rise and fall of a sudden. In hospitals, they were treated normally each 4 hours for check-ups where the nurses will check the temperature and the pulses from the patient. Blood sample need to be taken from the patient to confirm the diagnosis of the disease and identify the infectious parasites. For the blood sample to be taken, the ideal condition is when the body temperature of the patient is at the highest level. Experienced doctor will know the certain ideal time when to take the blood from the patient. However, the case is not the same with inexperienced doctors. What makes it worrying is that if they miss to take the blood when the temperature is at the highest level. What makes it even worst is that by missing the ideal time to draw the blood which will slower the diagnosis process, it will be vital to the patient's life. This is where the "Body temperature detection system for malaria fever patient using RFID and GSM network" comes to the picture. The patient will be receiving the quickest treatment as the system will send signals anytime the temperature exceeds the limits. The signals will then notify the nurses through alarm at nurse's station and send via SMS to the patient's doctor.

ACKNOWLEDGEMENTS

Firstly I would like to praise Allah the Almighty, which have helped and guided me in completing my Final Year Project 2 (FYP 2) in Universiti Teknologi PETRONAS. I also like to convey my deepest appreciation to my family, for their support, giving me motivation and advices, understanding, and their encouragement for me to be a better human.

My almost gratitude goes to my project supervisor, Puan Hanita Daud for her supervision and support throughout the process of the project. I would also like to express my full appreciation towards my co-supervisor, Mr Azizuddin Abd. Aziz; who have provided me with technical knowledge and ideas for this project to be successful.

In additional, my thanks also go to my fellow friends and individuals who were supportive throughout the project period. The support and encouragement from the people above will always be pleasant memory throughout my life.

Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

RFID	RADIO FREQUENCY IDENTIFICATION
GSM	GLOBAL SYSTEM FOR MOBILE
SMS	SHORT MESSAGING SERVICE
GUI	GRAPHIC USER INTERFACE
PC	PERSONAL COMPUTER
VB	VISUAL BASIC
PCMCIA	PERSONAL COMPUTER MEMORY CARD INTERNATIONAL ASSOCIATION
USB	UNIVERSAL SERIAL BUS

CHAPTER 1
INTRODUCTION

1.1 Background of Study

For this project, the system is basically a design for malaria fever patient at hospital in knowing the ideal time to take blood sample for diagnosis. Malaria fever patient is known to have irregularities in body temperature. The ideal time to take blood sample from them is when the temperature is at its highest level. By using the system, whenever the body temperature is exceeding the limit that has been set, the system will be activated. The system uses RFID and GSM network for the project to achieve its objectives.

1.1.1 Overview of RFID

Radio Frequency Identification (RFID) is a new wireless technology relying on storing and remotely retrieving data using RFID tags. [1] It has staggering capabilities and potential through implementation, monitoring, forecast and security. The main feature of RFID technology is its ability to identify, locate, track, and monitor people and objects between the tag and the reader.

[1] There are three basic components in RFID system, which is the tags, readers, and antenna. RFID tag contains at least two parts. One is an Integrated Circuit (IC) for storing and processing information, modulating and demodulating a radio frequency (RF) signal, and other specialized functions. The second is an antenna for receiving and transmitting the signal. As the IC used in RFID tags can hold a large amount of data, they can also include such information as serial numbers, medical records, etc.

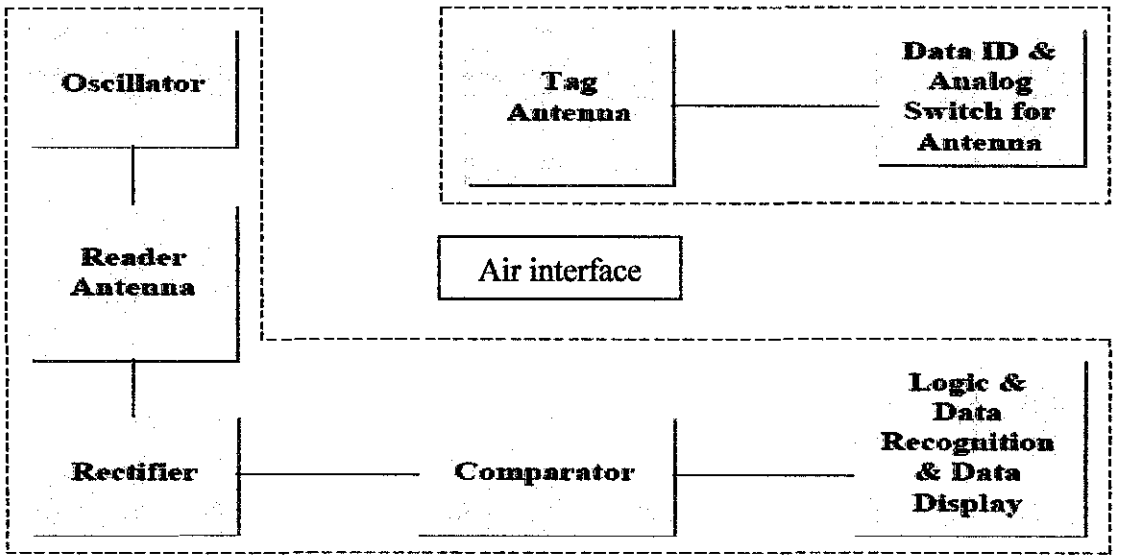


Figure 1: Basic block diagram of RFID

From Figure 1, [9] one can see that the RFID tag contains two sub circuits; a reader/receiver circuit and a tag/transmitter circuit. Initially the code that is desired to be sent across the reader is input into the DIP Switch on the tag, this data is then manipulated to control an analog switch in such a manner that the capacitor/inductive tank is switching on and off at a rate which corresponds to the data. When the tag is in the near field of the reader, it receives the signal and the Oscillator modulates the data, where the rectifier can change the incoming signal into a DC level which the comparator then determines if that DC level is above a certain threshold, and if it is, the output of the comparator is a one and if not, it is a zero. This data is then input into a serial in/parallel out shift register and then more logic determines the data code of the incoming signal and outputs it onto a seven segment LED.

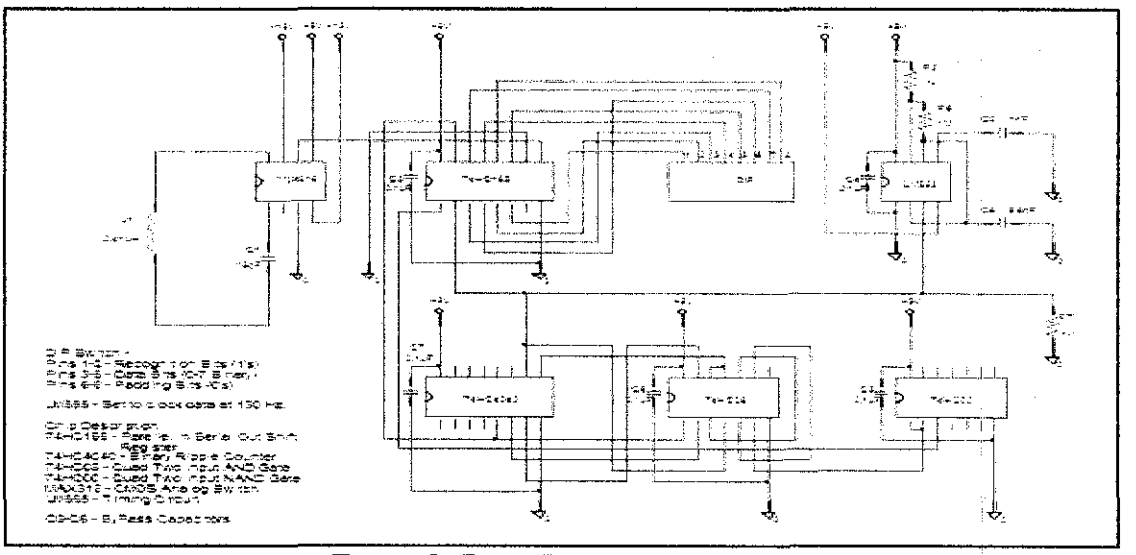


Figure 2: Basic RFID Tag Circuit Diagram

Figure 2 shows the basic RFID Tag Circuit Diagram. RFID tags come in three general varieties which are passive, active, and semi-passive. No internal power source required for passive tag, thus it will only respond when the reader transmits signal by backscattering process, a scientific term for reflecting through the mirror. Semi-passive and active tags require a power source, usually a miniature battery. Even though semi-passive tags contains internal power source, it shares the same concept with passive tag for transmitting signal back to reader; the backscattering process. The power source in semi-passive tag used just to store data and information.

Apart from passive and semi-passive, power source that contains in active RFID tags does not only can store much bigger data and information, it also can broadcast response signal to the reader. Communications from active tags to readers is typically much more reliable as there are only fewer errors than from passive tags due to the ability for active tags to conduct a "session" with the reader.

[2] The advantages of active RFID tags are:

i) Can be read up to 100 meters from the RFID reader.
ii) Battery life of up to 5 years.
iii) Transmit signals at higher power levels than passive tags, allowing them to be more robust in RF challenged environments.
iv) Highest data bandwidth.
v) May have sensors that can use electricity for power.
vi) Capable to perform independent monitoring and control.

[2] However, there are still disadvantages using active RFID tags such as:

i) Typically far more expensive than passive and semi-passive.
ii) Physically larger that can limit applications.
iii) Cannot function without battery, which limits the lifetime of the tag.
iv) Long term maintenance costs if the battery were to be replaced.

There are 2 conditions that can be performed in transmitting and receiving signal from RFID tags and readers. Firstly, there is Reader Talk First (RTF). RTF condition is suitable used for passive and semi-passive tags where the reader will first transmit the signal to the tag and the tag will respond by sending the information stored via backscattering process.

Second condition is Tag Talk First (TTF), where active tags are much more suitable. The condition will be started once the tag is triggered in certain circumstances and transmitting signal for the data information stored to be read by the reader. Usually, the TTF condition uses a special RFID tag which contains various specialized sensors, such as temperature, humidity, shock/vibration, light, and pressure. For example, the temperature sensor can been used to record the temperature profile during the transportation and storage of perishable goods. Figure below shows an active RFID tag with a temperature sensor.

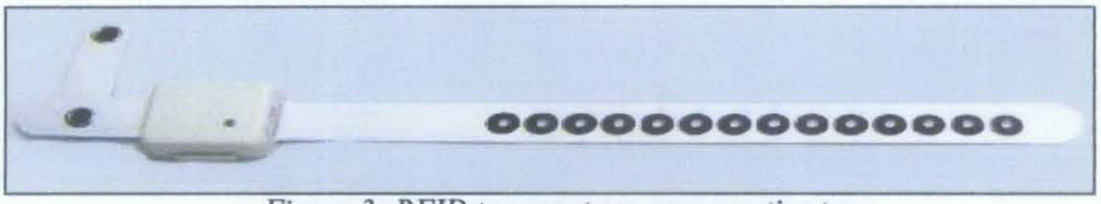


Figure 3: RFID temperature sensor active tag

[9] RFID reader is a device is used to interrogate RFID tags. An RFID reader typically contains a module (transmitter and receiver), a control unit and a coupling element (antenna). Figure 4 shows the basic RFID reader diagram.

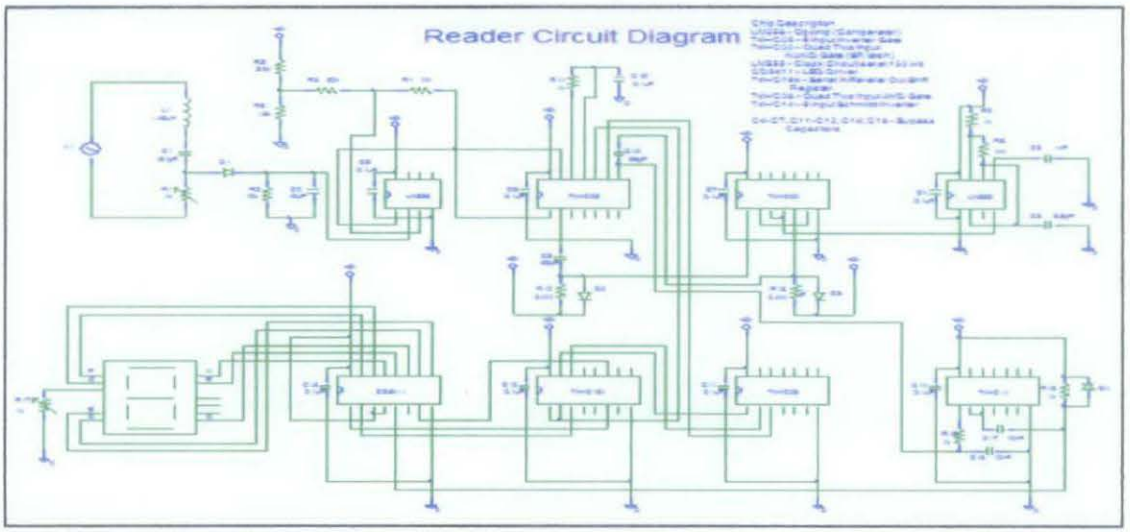


Figure 4: Basic RFID Reader Diagram

The reader has three main functions: energizing, demodulating and decoding. Energizing is a process when an RFID reader emits a low-power radio wave field which is used to power up the tags. Demodulation is when the reader extracts the original information-bearing signal from modulated carrier wave by RFID tags. Decoding is the process where turning the raw incoming data received from RFID tag into meaningful context information for further processing and subsequent actions.

[1] In addition, readers can be fitted with an additional interface that converts the radio waves returned from the RFID tag into a form that can then be passed on to another system, like a computer or any programmable logic controller. Anti-Collision algorithms permit the simultaneous reading of large numbers of tagged objects, while ensuring that each tag is read only once. RFID reader obtains the information of objects and surroundings through communication with RFID tag antennas. Reader and tags communicates using wireless connection. Figure 5 shows an RFID reader:

Figure 5: RFID reader

[9] The basic design of antenna were included by researching for formulas which calculate the inductance of a coil given the radius of the antenna, the radius of the wire used, and the number of turns. This equation is as follows:

$$L = N^2 \cdot R \cdot \mu_o \cdot \ln\left(\frac{R}{r}\right) \text{ (H)}$$

[9] The values that were used in this design are the same for the reader antenna and the tag antenna. The number of turns (N) is four, the radius (R) is four centimeters, the radius (r) is .405 millimeters, and the value μ_0 is $4\pi x10^{-7}$. Plugging these numbers into the above equation yields a value of 14.79 μH. After this was done it was necessary to find the value of the capacitor which will create a 13.56 MHz resonating circuit. To do this, the equation used is:

$$C = \frac{1}{(2\pi)^2 \cdot f_o^2 \cdot L} \text{ (F)}$$

[9] Plugging the respective values into this equation yields a value of 9.31 pF for each capacitor in the circuit. After these values were calculated, the range of the antennas was to be tested. Testing of the antennas showed that the range of the antennas was approximately six inches. The proper resonant frequency must be chosen so that the tag can be suitable for rectification on the reader side of the system.

$$C = f\lambda$$

Some RFID tags and readers perform in different frequencies. Through the equation used as shown above, with the value for speed of light (C) remains constant, the variables for this equation is frequency (f) and distance (λ). The relationship between f and λ is inversely proportional; where if value of f is high, value for λ will be lower and vice versa.

This project is designed with two actions performed when a condition occurs; which is the temperature level of the patient has reached its highest limit. The actions performed are notifications to patient's doctor via SMS (using the GSM modem), and alarm at the nurses' station been activated for immediate action while waiting for the doctor.

The temperature sensor along with the RFID active tag will be placed at the patient's armpit. Activation of the temperature sensor will trigger the RFID transmitter circuit located at the active tag to transmit the signal to the RFID reader located at the nurse's station. Upon receiving the signal, RFID reader which contains

receiver circuit will then sent signal of the triggered location to the PC and send the information via SMS to the doctor's patient.

1.1.2 Overview of GSM

Global System for Mobile Communication (GSM) modem is a wireless modem that works with GSM network. [3] Wireless modem behaves like a dial-up modem but the main difference between them is that dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

[3] A GSM modem can be an external device or a PC Card / PCMCIA Card. Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate. Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, we can:

i) Read, write and delete SMS messages.

ii) Send SMS messages.

iii) Monitor the signal strength.

iv) Monitor the charging status and level of the battery.

v) Read, write, and search phone book entries.

[3] The number of SMS messages that can be processed by a GSM modem per minute is very low, which is only about six to ten SMS messages per minute. Figure 6 shows the GSM Modem:

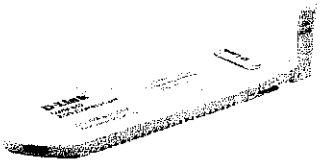


Figure 6: GSM Modem

1.2 Problem Statement

Malaria fever is usually a self-limited illness, and only supportive care is required. When a patient in or from a malarious area presents with fever, a blood smear should be prepared and examined to confirm the diagnosis and identify the species of infecting parasite.

Malaria fever patient suffers from fever spikes; where the body temperature rise and fall in a sudden. For doctors to examine the disease they have to see the pattern of the sudden rise of the body temperature and usually this occurs during nights. When the body temperature arises, usually during the nights, it is the perfect timing for doctors and nurses to take blood sample from the patients' body. However, there is still no solution as when the timing for taking blood sample is ideal as the body temperature will not be stabilizes.

Mostly in hospitals, the nurses will check the patient every 4 hours to take the patient's temperature and the blood temperature. The main concern is they will never know the ideal time to take the patient's blood when the temperature is at its highest level. For experienced doctors, they will know the ideal time, but for the inexperienced doctors, it shall be tough. So, by creating the new system using RFID technology, it can assures the patient will receive the quickest treatment as notifications via SMS to the patient's doctor and alarm at the nurse's station.

1.2.1 Project Significance

The project basically offers features of advancement in detecting the ideal time for taking blood sample for diagnosis to reduce highly risk of fatality. This system is based on integration and application of theories in engineering discipline especially in communications. The theories learnt in this project should be the base of learning experience and platform for better understanding on engineering principles.

The project is significance in creating a system that can automatically take temperature of the patient's fever at its peak (40°C). At that temperature, it is the time for the doctor to take the patient's blood sample because parasite or bacterium related to Malaria is active at that time. With that, the doctor can take note and monitor their patient's fever status.

Besides, it is significance in helping the nurse and also inexperience doctor in monitoring the patient's fever status. This is due to the hardness of the doctor to manually predict the patient's core body temperature to reach its peak (40° C). With this system, it can automatically detect it and send alarm to the nurse and the doctor. This will ease the nurse and inexperience doctor to take appropriate action when patient reach their fatal state.

1.3 Project Objectives

The objectives of the project are mainly focused on the wireless communication and sensor technologies. The project objectives are as follows:

i) To design a system that can notify the doctors and nurses the best possible time in taking blood samples for malaria fever patient.

ii) To reduce the risk of the patient in receiving late treatment from both of the nurses and doctors.

1.4 Scope of Study

In order to accomplish the objective, the scopes of study are:

i) Perform analysis of RFID tag and reader suitable for the project.

In achieving the purpose in understanding the process involves at hospital, the author should know how the current set up work. In designing this system, a good design should follow certain criteria. The most fundamental is whether the tag and the reader have a reasonable range of communication which will be vital as the reader might be located slightly far from the tag.

The second key criterion is whether the system able to operate reliably between the range of temperatures. The range should be enough for the reader to detect the rise of the temperature from normal condition.

ii) Perform analysis on GSM network.

For the doctors to receive notification via SMS, a GSM network will be linking up with the database system for which when the sensor triggered, it will automatically send SMS to the doctors. In designing the system, there are certain criteria needs to be fulfilled. The most fundamental is the SMS should be sent to doctors that were on duty only. Therefore, the on duty doctor must log in to the system to ensure that they will be given the notification whenever the system triggered. The second criterion would be whether the network system is able to operate reliably for bigger ranges. The range should be wider as the doctors will attend other patients which may put them in a greater range from the system.

iii) Perform analysis on Malaria fever condition.

The author must know the conditions of the disease to make sure that project will achieve its goals and objectives. To confirm diagnose on malaria fever patient, a blood sample must be taken for the doctors to identify the infectious parasites. The most fundamental criteria that need to be fulfilled are the degree of the body temperature of the patient. There will be a certain level of temperature where the certain parasites will be active and the blood should be taken during that time to confirm the diagnosis.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview of Malaria

Malaria is a mosquito-borne disease caused by a parasite. [10] Each year 350-500 million cases of malaria occur worldwide, and over one million people die, most of them young children in sub-Saharan Africa. Malaria is widespread in tropical and subtropical regions, including parts of the Americas, Asia, and sub-Saharan Africa. [11] Malaria is caused by a protozoan parasite of the genus *Plasmodium* and is transmitted to animals by the bite of a female mosquito of the genus *Anopheles*. Malaria can cause many symptoms, the most common of which include fever, fatigue, chills, and many other flu-like symptoms. Though there is no vaccine to protect against this disease, many preventative and treatment methods are available.

2.1.1 Geographic distribution of Malaria in the world

Where malaria is found depends mainly on climatic factors such as temperature, humidity, and rainfalls. Malaria is transmitted in tropical and subtropical areas, where:

- *Anopheles* mosquitoes can survive and multiply
- Malaria parasites can complete their growth cycle in the mosquitoes ("extrinsic incubation period").

Temperature is particularly critical. For example, at [11] temperatures below 20°C (68°F), *Plasmodium falciparum* (which causes severe malaria) cannot complete its growth cycle in the *Anopheles* mosquito, and thus cannot be transmitted.

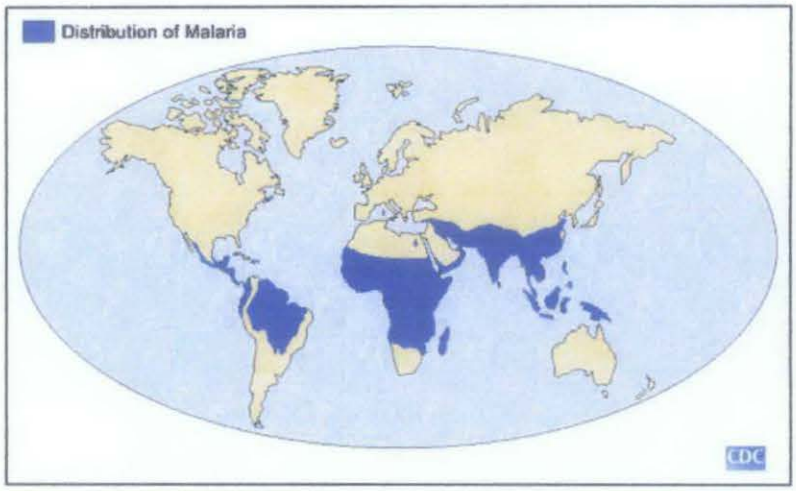


Figure 7: Geographic Distribution of Malaria

[11] Even within tropical and subtropical areas, transmission will not occur:

- At high altitudes
- During cooler seasons in some areas
- In deserts (excluding the oases)
- In some islands in the Pacific Ocean, which have no local *Anopheles* species capable of transmitting malaria
- In some countries where transmission has been interrupted through successful eradication.

Generally, in warmer regions closer to the equator:

- Transmission will be more intense
- Malaria is transmitted year-round
- *P. falciparum* predominates.

[10] The highest transmission is found in Africa South of the Sahara. In cooler regions, transmission will be less intense and more seasonal. There, *P. vivax* might be more prevalent because it is more tolerant of lower ambient temperatures.

According to WHO, about 40% of the world's population, mostly those living in the poorest countries, are at risk of malaria. Of these 2.5 billion people at risk, more than 500 million people become severely ill with malaria every year and more than 1 million people die from the effects of the disease.

[11] Malaria occurs in over 100 countries and territories. 41% of the world's population live in areas where malaria is transmitted (e.g., parts of Africa, Asia, the Middle East, Central and South America, Hispaniola, and Oceania). In 2002, malaria was the fourth cause of death in children in developing countries, after perinatal conditions (conditions occurring around the time of birth), lower respiratory infections (pneumonias), and diarrheal diseases. Malaria caused 10.7% of all children's deaths in developing countries. Not all people go to hospitals when sick or having a baby, and many die at home. Thus the true numbers of death and disease caused by malaria are likely much higher. (WHO 2007).

Table 1: Number of malaria-endemic countries, 1990–2003

Region	Subregion	Total no. of countries	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003[a]
Africa	Central	8	6	6	6	6	6	7	7	6	7	4	4	4	2	3
	East	12	5	5	7	9	9	10	9	7	10	11	10	9	9	8
	North	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2
	Southern	11	7	6	7	8	8	10	9	8	10	11	11	11	10	6
	West	16	13	13	12	13	14	16	15	16	14	14	13	10	7	3
Asia	Central Asia & Transcaucasia	7	7	7	7	7	7	7	7	7	7	7	7	7	7	6
	Eastern Mediterranean	9	9	9	8	8	9	8	9	9	8	9	9	8	9	9
	South-East Asia	10[b]	8	8	8	8	8	8	8	8	10	8	10	10	10	9
	Western Pacific	10	10	10	10	10	10	10	10	10	10	10	10	10	10	9
The Americas	Central America & Caribbean	10	10	10	10	10	10	9	10	9	10	10	10	10	9	9
	South America	11	11	11	11	10	11	11	10	11	11	11	11	11	11	11
Total		107	90	89	90	93	96	100	98	95	101	99	99	94	88	76

[a] As a result of a general delay in the receipt of national reported case rates at WHO headquarters, the number of countries reporting in 2003 is not yet complete.

[b] The number of countries increased from 9 to 10 with the establishment of Timor-Leste in 1999.

Table 2: Reported malaria cases in Malaysia, 2000-2003

Reported malaria cases by selected subnational area in Malaysia

15 of 15 areas	2000	2001	2002	2003	%
Sarawak	3011	3145	2496	2615	41
Sabah	5776	6050	5096	1770	28
Pahang	1301	1544 1	1563	850	13
Johor	710	671	579	284	4
Perak	852	470	280	276	4
Selangor	271	172	159	119	2
Pulau Pinang	209	197	76	106	2
Kelantan	386	184	333	99	2
Kedah	12	26	82	92	1
Terengganu	94	76	140	47	1
Negeri Sembilan	37	205	180	45	1
W.P.Kuala Lumpur	27	20	15	20	<1
W.P.Labuan				7	<1
Melaka	18	15	16	7	<1
Perlis	1	5	4	1	<1

Total cases : 12,075 12,777 11,019 6,338

2.2 Overview of RFID Applications

2.2.1 RFID Temperature/ID tags spots bird flu (6)

Digital Angel Corporation had introduce its patented 'Bio-Thermo' temperature-sensing, implantable RFID chip and identity system to international poultry/bird markets with an initial geographic emphasis in Asia - with a view to helping poultry breeders spot early signs of disease such as Avian Flu. This company for now has focused its implantable RFID temperature sensing system on the pets and horses.

The identity system includes the Bio-Thermo chips and related ID and temperature-sensing RFID scanners that aim to provide early identification of temperature increases in individual birds. With this, it may in turn help with the identification and control of outbreaks of avian diseases.

With early detection of temperature increases, identifying and controlling disease outbreaks become important; such monitoring could help farmers avoid the complete destruction of their stock by identifying, quarantining and monitoring suspected cases at an individual level.

Besides using it for the birds, this Bio-Thermo is also being used in pets. The chip will be put into the pet's body and be registered specifically to the owner. The benefit is that it can take accurate reading of the pet's core body temperature.

2.2.2 Monitoring and traceability of hemoderivates (7)

There is a system used by hospitals to secure and monitor the temperature of their blood bank. Whenever the temperature of the blood bank is over its limit, the system will send alarm for the hospital to take action. It is also used in blood bag that they wanted to be transfer to somewhere else.

This system provides product quality assurance before the blood bags are used by discarding the ones that turn out to be defective because of insufficient conservation standards. So, the design of this new system is for hemoderivate product storage and production technology. In effect, this will avoid many blood transfusion risks.

2.2.4 RFID Pill Monitors Body Temperature (8)

Researchers in The Netherlands were able to monitor the body temperature by swallowed an RFID-based temperature sensor that measured their internal temperature. This will helped researchers identify potential health issues. This RFID technology has been used by volunteer participants at the world's largest marching event in the annual Four Days Marches of Nijmegen. With this pill, the researcher can monitor the marcher's core body temperatures so that they will not exceed recommended levels and prevent them from exhaustion or overheating.

Using complex event processing (CEP) technology, researchers were able to monitor and record the temperatures via a signal transmitted every ten seconds from the RFID "pill" to a receiving device in the volunteer's backpack.

That data was then transmitted via Bluetooth to a GPS-enabled mobile phone to the operations center handled by the researcher themselves. CEP is an event processing technology that allows an application to analyze multiple streams of event data, and then react to those conditions quickly. The CEP platform processed and analyzed the temperature data in real time. If a volunteer's body temperature was too high, officials could alert them to either rest or rehydrate using SMS text messaging, calling them on the mobile phone, or by alerting the onsite medical team to take action if needed.

2.3 RFID Frequencies

Table 3 shows the RFID operating frequencies and all associated characteristics of it. Each has advantages and disadvantages. The author chose ultra high frequency (UHF) band which suits the best to this project.

Table 3 : RFID operating frequencies and associated characteristics

Band	**LF Low frequency**	**HF High frequency**	**UHF Ultra high frequency**	**Microwave**
Frequency	30–300kHz	3–30MHz	300 MHz–3GHz	2–30 GHz
Typical RFID Frequencies	125–134 kHz	13.56 MHz	433 MHz or 865 – 956MHz 2.45 GHz	2.45 GHz
Approximate read range	less than 0.5 metre	Up to 1.5 metres	433 MHz = up to 100 metres 865-956 MHz = 0.5 to 5 metres	Up to 10m
Typical data transfer rate	less than 1 kilobit per second (kbit/s)	Approximately 25 kbit/s	433–956 = 30 kbit/s 2.45 = 100 kbit/s	Up to 100 kbit/s
Characteristics	Short-range, low data transfer rate, penetrates water but not metal.	Higher ranges, reasonable data rate (similar to GSM phone), penetrates water but not metal.	Long ranges, high data transfer rate, concurrent read of <100 items, cannot penetrate water or metals	Long range, high data transfer rate, cannot penetrate water or metal
Typical use	Animal ID Car immobiliser	Smart Labels Contact-less travel cards Access & Security	Specialist animal tracking Logistics	Moving vehicle toll

CHAPTER 3

METHODOLOGY/PROJECT WORK

Figure 8 shows the flow of the author's project.

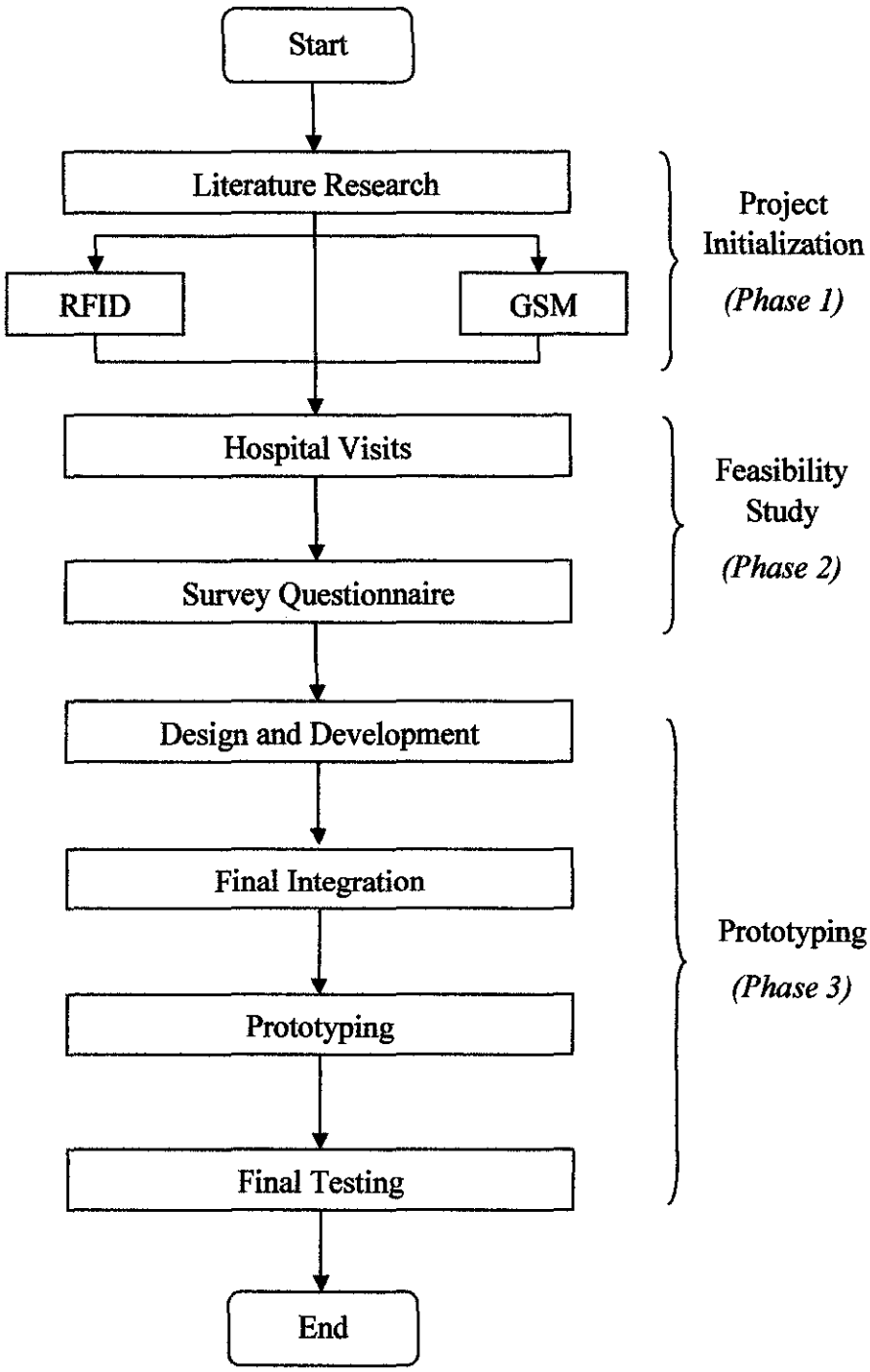


Figure 8: Flowchart of the project

3.1 Project Initialization (Phase 1)

Project Initialization is the early stage of the project development process. At this stage, firm understanding and planning on the project is seriously considered. A lot of research regarding the project using RFID technology and its applications has been held for the feasibility of the project. Research also was held for GSM modem technology for SMS notification.

3.1.1 Analysis of RFID frequencies

Radio-frequency (RF) signals are typically sinusoidal or nearly so that is, the voltage or field is a smooth, periodic function of time. The number of times the signal repeats itself per second, the frequency, varies widely in differing RFID systems. Frequency is measured in Hertz (Hz): one Hertz is one cycle per second. KHz= 1000's of Hz; MHz = millions of Hz. Several issues are involved in choosing a frequency of operation. Firstly, type of coupling, whether inductive or radiative coupling will be employed. The distinction is closely related to the side of the antennas to be used relative to the wavelength. Next issue in selection of frequency bands is the allocation of frequencies by regulatory authorities. In essentially every country in the world, the government either directly regulates the use of the radio spectrum, or delegates that authority to related organizations. In Malaysia, Malaysian Communication and Multimedia Commission regulates the frequencies that radios are allowed to radiate, the power levels they can use, and other more technical aspects of their operation. Finally, changes in operating frequency affect the propagation characteristics of the resulting radiated fields. Lower frequencies diffract more readily around obstacles, but couple less well to small antennas.

3.2 Feasibility Study (Phase 2)

The feasibility studies phase were started by conducting a hospital visits for further research about the disease and the connection with the author's project. The visits shall give an advantage for the author as there will be professional opinion given by doctors from the hospital regarding the project. Survey questionnaire will also be held to be distributed for the feasibility of this project.

3.2.1 Hospital Visits

A visit has been made to Hospital Sungai Buloh located at Selangor for the author to conduct a survey about the malaria fever syndrome and also the authorities to perform tests on the patients for the success of the project. The meeting with Dr. Khalid Ibrahim, the director of Hospital Sungai Buloh is such a pleasant experience as he gave the author his professional opinion about the project and lots of advice given to see the project succeeded.

3.2.2 Survey Questionnaire

A survey has been conducted to determine the feasibility of the project. The questionnaire is targeted for all as the disease can be infected at all types of ages. Through this survey, the author will calculate the percentage of the neediness for this project to be successful and to be used for future development. A feedback/suggestions section of the questionnaire was also included to improve the next questionnaire if another survey is conducted. (Please refer Appendix B). Results of the survey will be shown in Chapter 4: Result and Discussion section.

3.3 Prototyping (Phase 3)

Prototyping is a phase where the task of design and build based on the available and costless parts and components. Final integration will be focusing more on the interfacing between devices, hardware and system of the project. In the final testing stage, the combination of all the main features is tested together with the software application. Fine tuning will be done to ensure the smoothness of the project.

3.3.1 Design and Development

A system; known as BodyTemp Detaxx has been designed by the author by using Visual Basic as its Graphical User Interface (GUI). The author also decided to use MySQL as its database to store all the information from the system. Information regarding Visual Basic was obtained from written sources, such as books that provide guidelines to use Visual Basic. The development of the system is shown in the results and discussion section.

3.3.2 Final Integration

BodyTemp Detaxx will then be integrated to the hardware that will be used for the author's project. A lot of modification needs to be done as there were compatible issues regarding the GUI used that the hardware does not support. Integration between GSM Modem and also the system is to be made as the system will send SMS straight to the doctor's phone after the hardware detects that the temperature exceeds the limit. Figure 9 shows the steps taken in developing the body temperature detection system development using programming software.

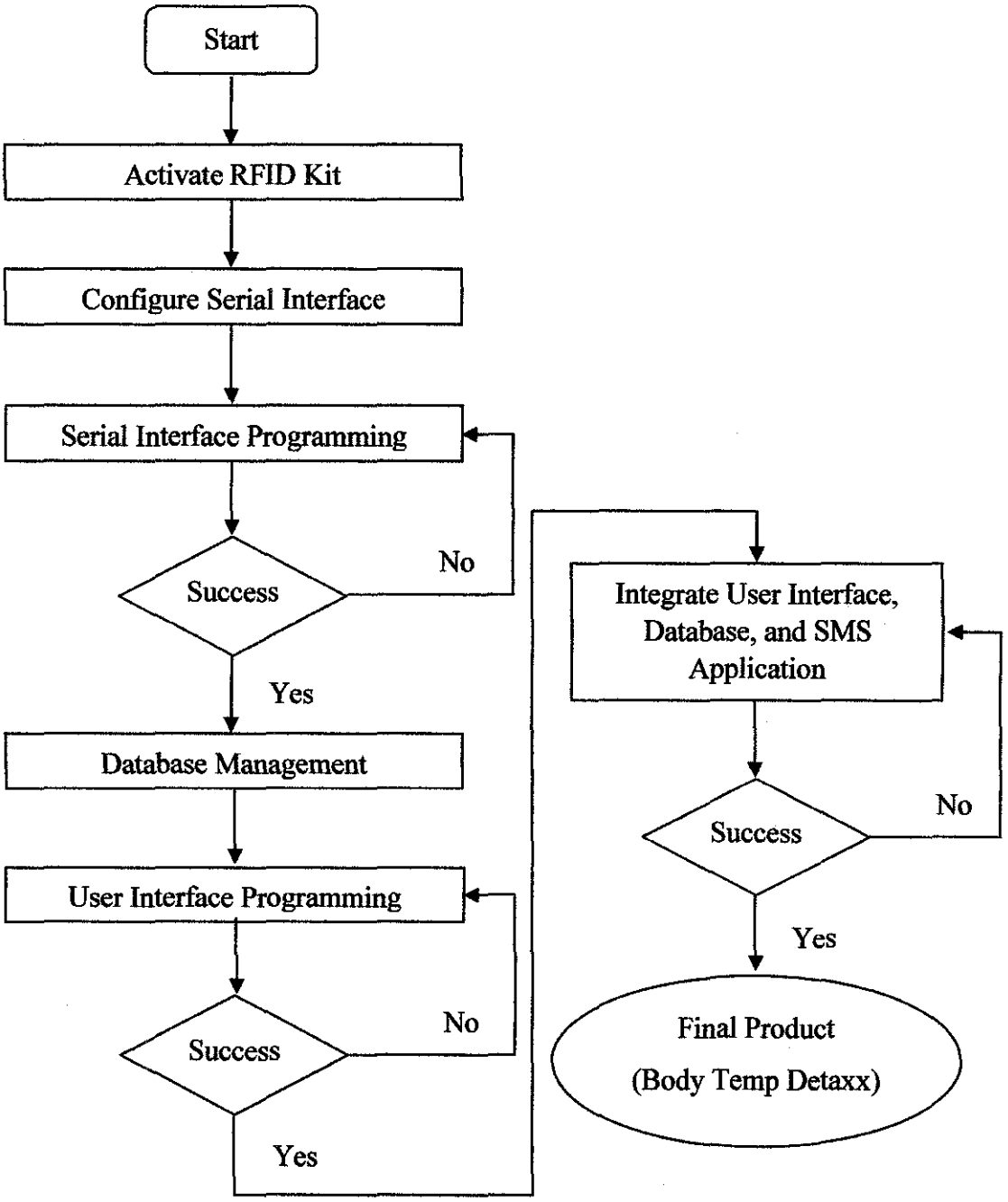


Figure 9: Flowchart of RFID Body Temperature Detection System

3.3.3 Prototyping

Prototyping were done after the final integration between the system and the hardware stage. The prototype which includes all the hardware that will be used will be tested at the next stage, final testing.

3.3.4 Final Testing

The final testing will be made by replacing actual malaria fever patient with a device that can control temperature. This stage is to make sure that the system functions well before the actual testing will be done at Hospital Sungai Buloh.

3.4 Hardware tools

3.4.1 Active Wave RFID Kit

Figure 10: Active Wave RFID Kit

Figure 10 shows the hardware tools that will be used for this project; manufactured by Active Wave.

RFID Kit Components:

i) 1 Active Wave Reader and Power Supply

ii) 1 Active Wave Reader RS-232 Connector Cable

iii) 1 Active Wave RJ-45 Cable

iv) 6 Active Wave Temperature Sensor Tags

v) 2 Active Wave Card Tags

vi) 2 Active Wave Jumbo Tags

vii) 1 Programming Station Software Application

3.4.2 Specification

i) Tags - Active Wave tag model that are available, transmitting at either 927 MHz, 916 MHz, or 868 MHz. All Active Wave tags have anti-collision circuitry that assures each tag's information is received when more than one tag is transmitting. Tags may be electronically enabled or disabled, so they can be 'seen' or 'unseen' by Active Wave Readers. On-board temperature sensors also were included in the tags to fit the project.

ii) Readers - Readers interface the Host applications to the rest of the Active Wave system. Reader transmits data at 433 MHz and receives data at 927 MHz, 916 MHz, or 868 MHz. Readers communicate to the Host computer via an RS-232 cable or via an Ethernet network connection. Readers are used to read the tags and transmit the received data to the Host computer. Readers also used to enable, disable, wake up and program the tags. The reason reader have different frequency with the tags is to prevent collision in retrieving data.

3.5 Basic Active Wave System

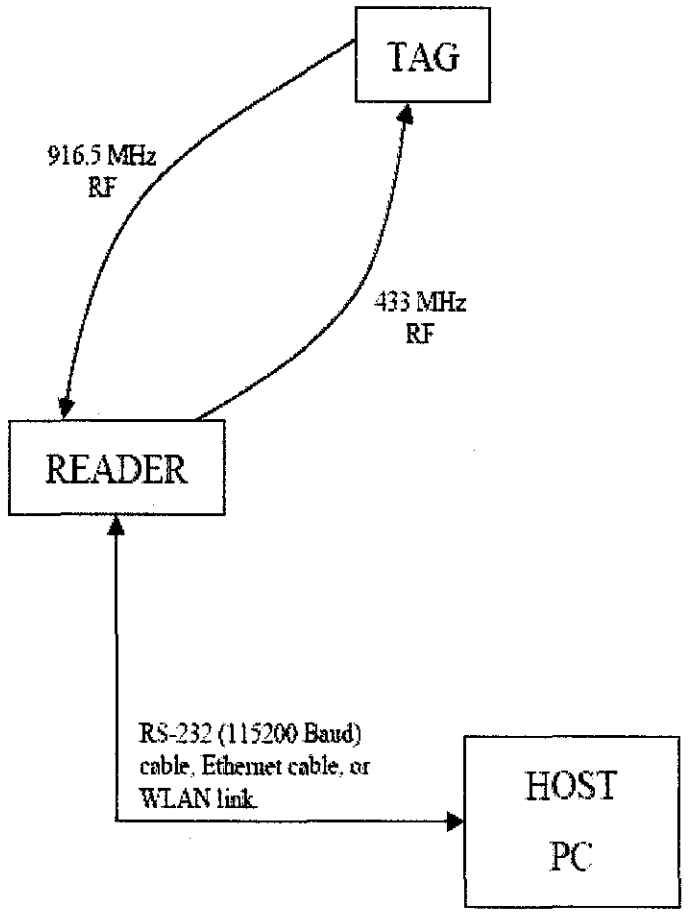


Figure 11: System flow of Active Wave

The basic Active Wave System consists only three components – the Host (PC running any Active Wave application), the Reader and at least one tag. Configuration of the Reader and tag are done via either an RS-232 connection or Ethernet connection. All monitoring and tracking of the tag are done via the same connection from the Host to the Reader.

3.6 Project Overview

The flow of the project is shown in Figure 12:

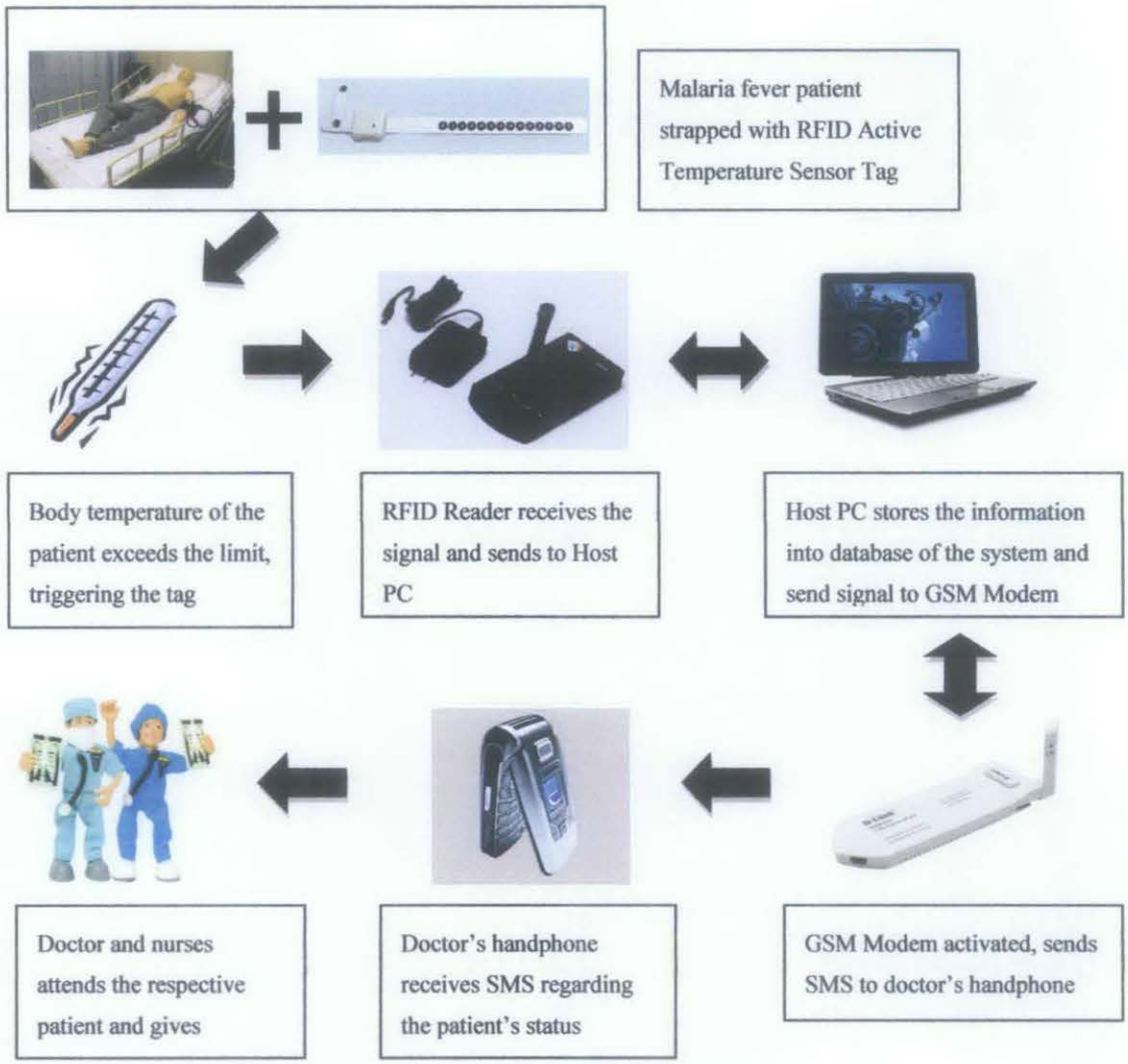


Figure 12: Project Overview

Firstly, the patient will be strapped with the tag at his/her armpit to determine the body temperature. The limit for the temperature to exceed is 40°C. Once the system triggered, the tag will send signal to the reader. The signal then will be sent to Host PC to store the information in the database. The temperature that has been stored at the database of the system at Host PC can be viewed through BodyTemp Detaxx. It will be saved as history for the doctors to analysis the trend of the patient's temperature. The system will be updating the temperature once every 1 minute.

Secondly, the signal will be sent to GSM Modem for it to be activated. SMS will be sent to doctor's handphone for him to be notified regarding the patient's status. The patient will be then receiving treatment from both of the nurses and doctor.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Survey Questionnaire

A survey has been conducted to determine the feasibility of this project. The survey was handed to 30 people of all kind of ages with 18 from 20-30 years group of age, 8 from 30-40 years, 3 from 10-20 years, and 1 from 50 years and above. Results from the survey by percentage of 30 people can be seen at Appendix B. The survey is useful for this project as this will determine the neediness of the public for this project to be successful and to be used for future development.

Section A (II) is the part in survey to determine the knowledge and experience of the respondents regarding malaria fever. 40% from the total respondent answers 'Yes', 43.33% answers 'No' while 16.67% is not sure. Majority of them know about malaria fever, but not for the conditions of the patients.

Section B is the part in survey to determine the criteria in selecting the best intensive care for malaria fever patient. Section B can be summarized that all of the respondents agree that a system with automatic updates is required for the nurses to keep track with the patients' body temperature.

Section C is the most important part in the survey which is to determine the feasibility of the project. 73.33% of the respondents agree to receive treatment at hospitals that offer "Body temperature detection system for malaria fever patient using RFID and GSM network".

Based on the results from the questionnaire, we can summarize that the project is feasible to be done as it is important for a new system with automatic updates required for the patients' safety.

4.2 Graphical User Interface (GUI) using Visual Basic

This project requires Graphical User Interface (GUI) for the nurses to fill in the details of the patients and save the required information. Visual Basic provides an easy GUI design using the Form Layout. Figure 13 shows the Login Page for the system. User must enter their username and password either as Administrator or as User. Administrator is the person involved in the process of adding information regarding the patient. Administrator is restricted to only nurses, while User is for doctors and also nurses that they can only view the patient status.

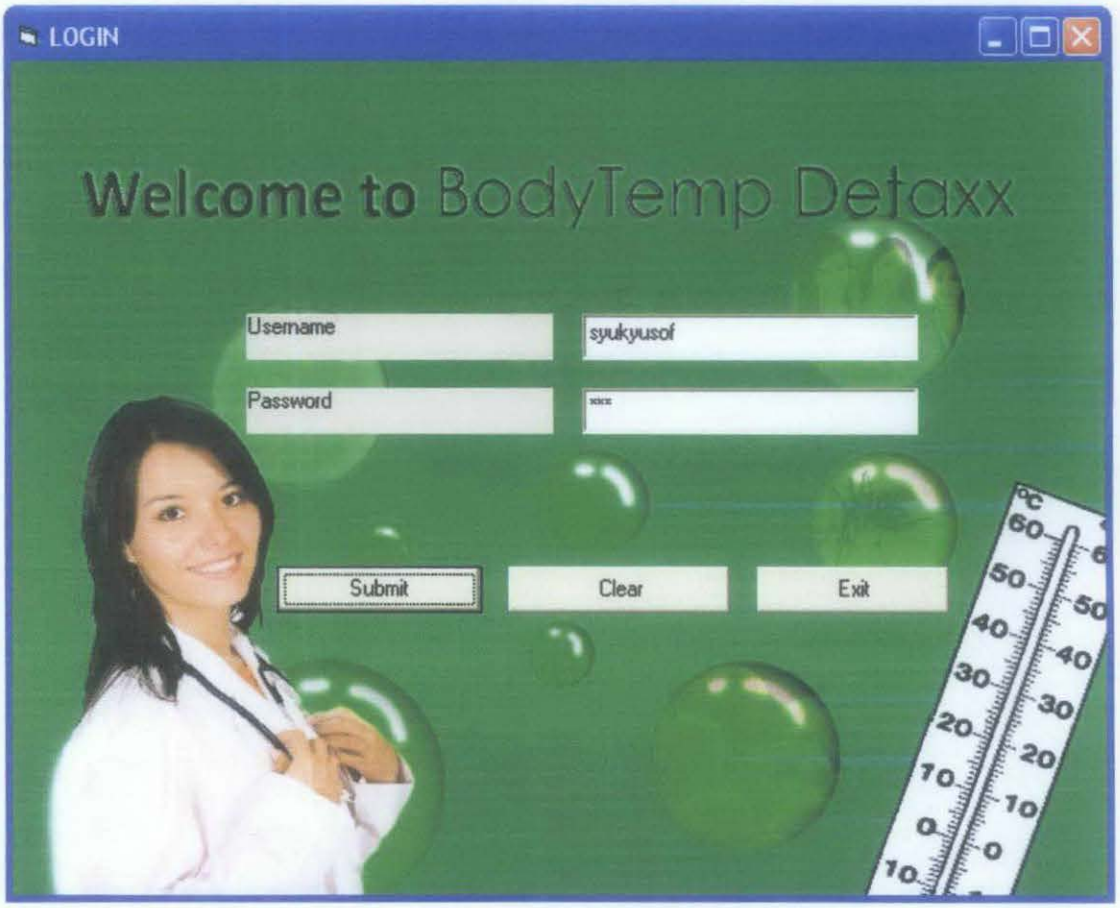


Figure 13: Login page

Figure 14: Main Menu

Figure 14 shows the Main Menu once the nurses have insert their ID and password in the Login Page. In this page, only Administrators can access System Administration; whereby only nurses are allowed to add information regarding the patients and also new user. The 'Start' option is to connect the system with the hardware. Exit section will bring the user to Login Page back.

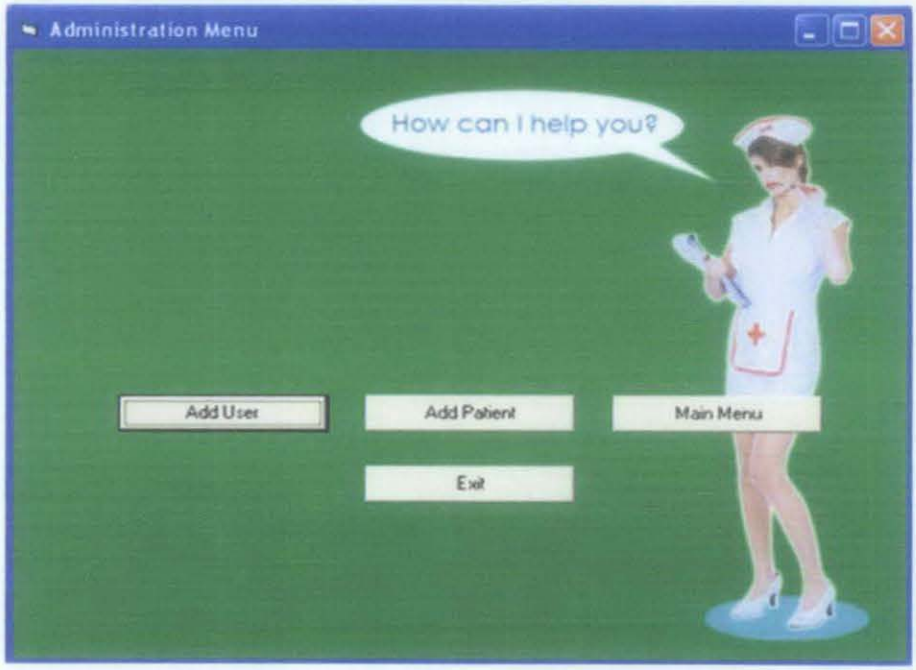


Figure 15: Administration Menu

Figure 15 shows the Administration Menu page for the nurses to insert details for new patient. For doctors to use the system, nurses will add their personal details in Add User. Doctors have no access to insert details about patients as they can only view the patients' status from the system. Main Menu section will bring back the user to Main Menu, while Exit section is for the user to quit the system.

Figure 16: Administration Page

Figure 16 shows the Add User section from previous Administration Menu Page. The information required will include the user's name, i/c no, position, date of birth, username, and the password. The submit section is where the information will be stored at the database of the system as shown in Figure 17.

Figure 17: Database for Add User section

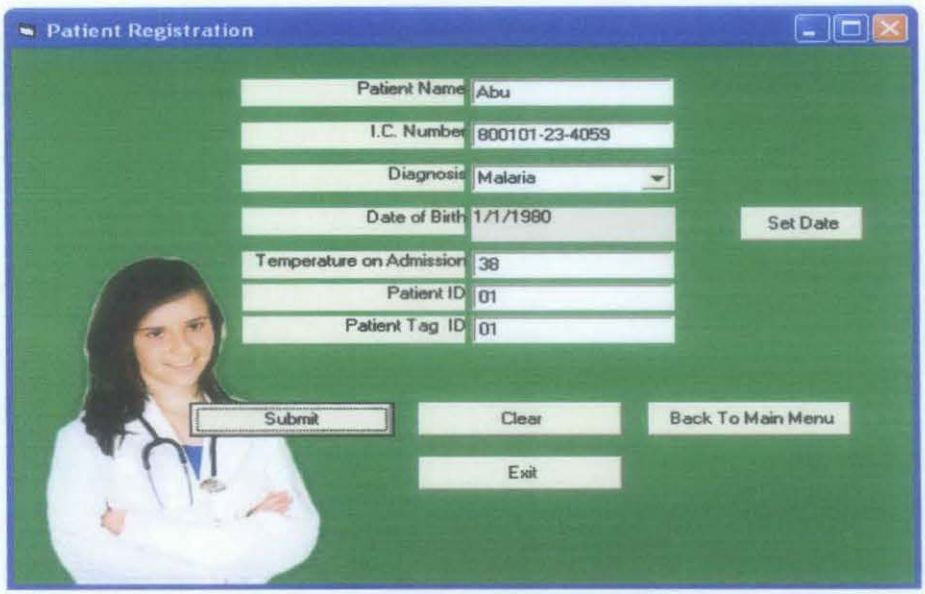


Figure 18: Patient Registration Page

Figure 18 shows the Patient Registration Page for Add Patient section from previous Administration Page. The information required will include the patient's name, i/c no, diagnosis of disease, date of birth, temperature on admission, patient ID and also patient tag ID that the patient currently occupying. The submit section is where the information will be stored at the database of the system as shown in Figure 19.

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

←T→	name	ic_num	tagid_patid	dateofbirth	dateofadmin	diagnosis	temperature	timeofadmin	age
☐ ✎ X	Aurangzeb	880828-05-2859	204	1988-08-28 23:03:58	2009-06-01 23:04:13	malaria	35	2009-06-01 23:04:37	17
☐ ✎ X	Abu b. Ali	800101-23-4059	201	1980-01-01 12:23:23	2009-06-04 12:24:50	malaria	38	2009-06-04 12:24:50	24

Check All / Uncheck All With selected: ✎ X

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Figure 19: Database for Add Patient section

For the system to begin, the 'Start' section from previous Main Menu page is shown in Figure 20. This section is to activate the system by connecting it with the hardware. Once the connection is done, the 'OFF' signal will be replaced by the 'ON' signal.

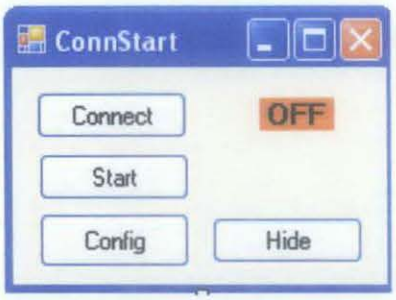


Figure 20: Start Option

When the nurses or doctors want to gather about their patient's information, they just have to click on 'View Patient Records' on the 'Main Menu' and form as in Figure 21 will come out. The nurses or the doctors just have to enter Patient ID and this interface will show all of the patient's information gathered from related database.

Figure 21: Patient Records section

For example, we want to check the temperature record for patient ID 201, once the 'search' button were pressed, Figure 22 will come out and show the record of the patient. The 'confirm details' button is to show the previous temperature that have been recorded by the system.

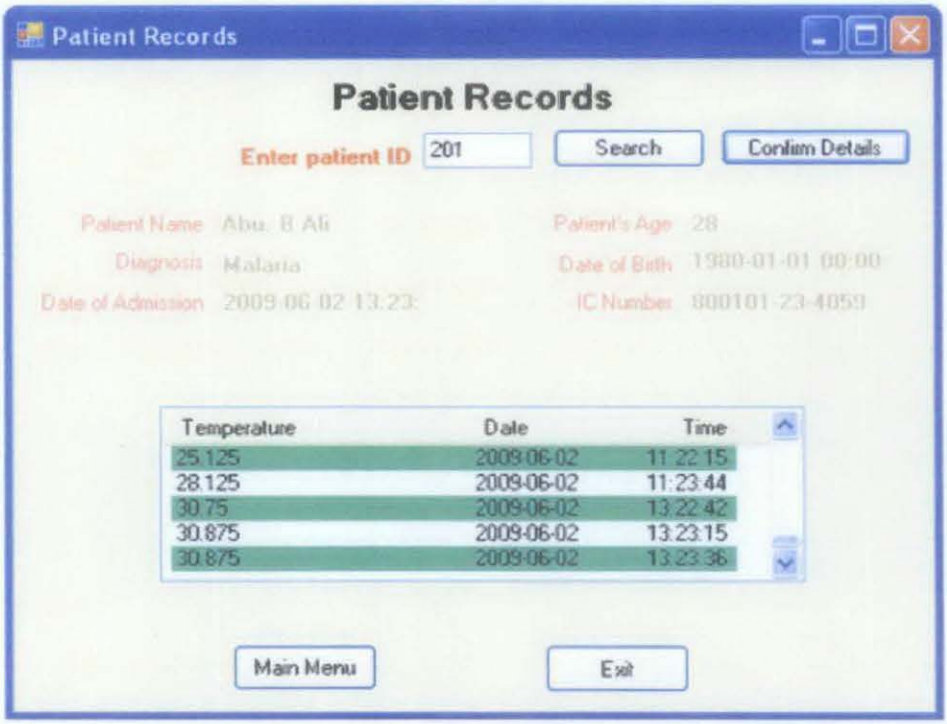


Figure 22: Temperature value for Patient 201

The temperature value for patient ID 201 were gained from database as when the system were activated, the history of the patient's temperature will be stored for the doctor to analysis the patient's temperature pattern. The value will be stored every 1 minute. Figure 23 shows the database for temperature values.

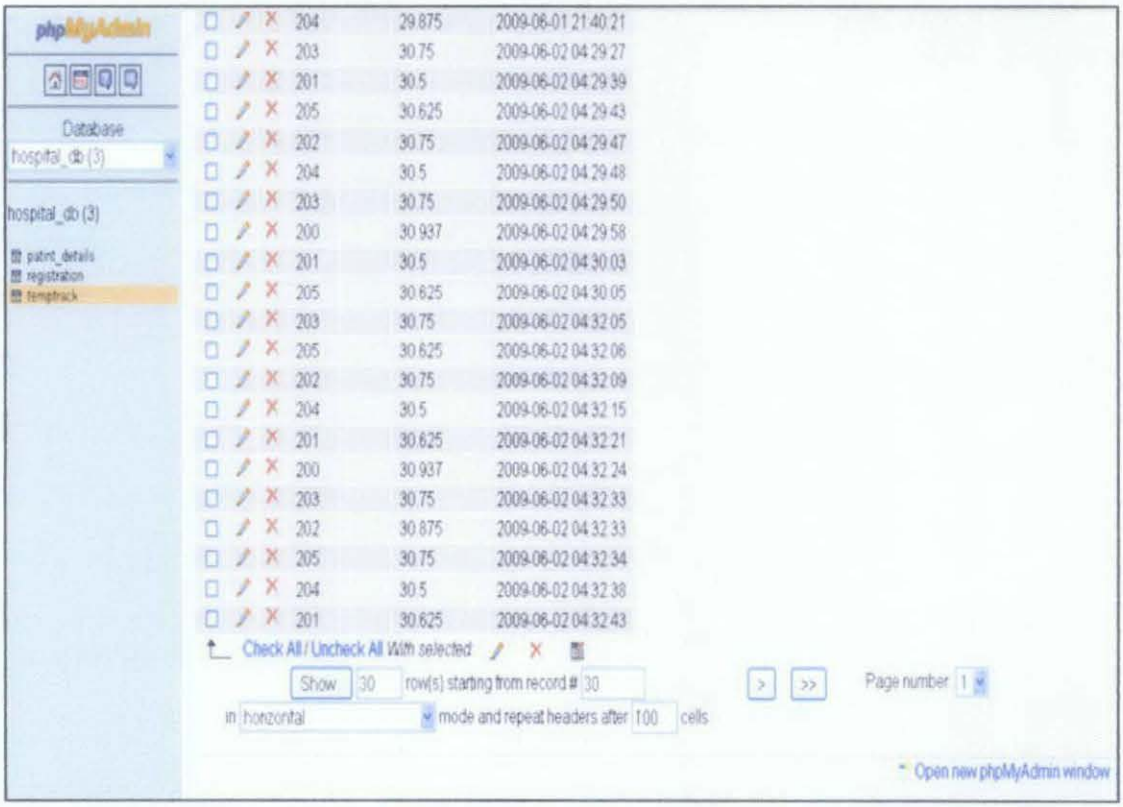


Figure 23: Database for temperature values

4.3 Interface between Database and GSM Modem

The GSM modem will be triggered to send message to the doctor after the RFID sensed temperature of the patient more than 40°C. Whenever there is an alert about any patient who reached high body temperature, the form as in Figure 24 will come out. Then the nurse will notify with it and when they click 'Message the Doctor' button, it will automatically switch on the GSM modem, send SMS and notify the doctor about the patient who needs them.

Figure 24: Alert Form

4.4 Discussion

As the project is based on communication system with largely programming language involved, it has tested the authors' capability and knowledge to its limits for this project to be done. Lots of problems arise during the completion of this project. Among the problems are the purchase of the hardware is done late due its cost. There were lots of procedures to follow to achieve the significant budget for the hardware to be purchased. Before the hardware were been purchased, the author has done the system by using database Microsoft Access. However, due to compatible issues, the author has to change the whole system for the system to be using MySQL as its database. This problem has really taken the author to its limit but the problems were settled before the final presentation occurs. These challenges were faced by the author in order to finish the project before its due date and the author have succeeded to achieve that feat. The project leaves lots of memories to the author through the hardness that been faced and handled before its completion.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The objective of this project is to design a system that can notify the doctors and nurses the best possible time in taking blood samples for malaria fever patient by notification via SMS to reduce the risk of the patient in receiving late treatment. The objective of the project has been achieved as the working prototype that can detect the body temperature of malaria fever patient has been developed. Notifications via SMS to the doctor and alarm at the nurse station should be adequate to prevent further complications if late treatment were given.

5.2 Recommendation

There is no restriction in improvement for this project. The recommendation for better advancement will benefit the system to realize with real applications in wider fields. Thus, the recommendations are as follows:

- To improve the notification system that can accommodate the consumer need by using other application offered by GSM modem such as voice calls and wireless internet.
- Implementations of other RFID sensor tags that can be detect all the necessary information of the patient other than body temperature, such as blood pressure. This information should help the nurses and the doctors to act faster when all the information were given.

REFERENCES

[1] Linda Castro & Samuel Fosso Wamba, "An Inside Look at RFID Technology", *Journal of Technology Management & Innovation, 2007, Volume 2 Issue 1, page 128.*

[2] Radio Frequency Identification: An Introduction for Library Professionals. Alan Butters. Australasian Public Libraries v19.n4 (2006) pp.2164–174 RFID Technology

[3] Byron S. Gottfried, *Schaum Outlines: Visual Basic, Singapore*: McGraw-Hill, 2001.

[4] Joachim Tisal, *The GSM Network GPRS Evolution: One Step Towards UMT,* Chichester, England: John Wiley & Sons, 2001.

[5] Michael Ekedahl & William Newman, *Programming with Microsoft Visual Basic 6.0: An Object-Oriented Approach,* United States of America: Course Technology, 1999.

[6] UsingRFID.com. *UsingRFID.com Web site.* Digital Angle Corparation. [Cited: February 8, 2009.]

[7] RFID Convocation. *RFID Convocation Web site.* [Cited: February 08, 2009.]

[8] RFID update - The RFID Industry Daily. *RFID update - The RFID Industry Daily Web site.* [Cited: March 3, 2009.]

[9] Eagles Nest, The (2002, September 27th). *RFID: The Early Years 1980-1990.*

[10] Snow RW, Guerra CA, Noor AM, Myint HY, Hay SI (2005). *"The global distribution of clinical episodes of Plasmodium falciparum malaria". Nature* **434** (7030): 214–7. doi:10.1038/nature03342. PMID 15759000.

[11] "Malaria: Disease Impacts and Long-Run Income Differences" Institute for the Study of Labor. Retrieved on 2008-12-10.

APPENDICES

Appendix A: Gantt Chart

Appendix B: Survey Questionnaire

Appendix C: Database Tables

Appendix D: Visual Basic Source Code

APPENDIX A

GANTT CHART FOR FYP1

No.	Detail/ Week	1	2	3	4	5	6	7	8	9	10	11	12	12	13	14
1	Selection of Project Topic	■	■									Mid-semester break				
2	Preliminary Research Work		■	■	●											
3	Submission of Preliminary Report															
4	Project Work					■	■	■	●							
	- Hospital Visits									●						
	- Survey Questionnaire									●						
5	Submission of Progress Report										●					
6	Seminar (compulsory)															
7	Project work continues									■	■		■	■	■	■
8	Submission of Interim Report Final Draft														●	●
9	Oral Presentation															

● Suggested milestone

■ Process

GANTT CHART FOR FYP2

No.	Detail/ Week	15	16	17	18	19	20	21	22	23	24	Mid-semester break	26	27	28	29
1	Design and Development of Database	■	■	■	■	■	■	■	■	■	■					
2	Submission of Progress Report 1															
3	Submission of Progress Report 2												●			
4	Integration Database with Hardware												■	■		
5	Seminar (compulsory)														●	
6	Prototyping														■	■
7	Submission of Final Dissertation Report															●
8	Submission of Technical Report															
9	Oral Presentation		●													

● Suggested milestone

■ Process

APPENDIX B

SURVEY QUESTIONNAIRE

Body Temperature Detection System for Malaria Fever Patient using RFID and GSM network

Malaria fever patients are known to suffer from fever spikes, where the body temperature rise and fall of a sudden. In hospitals, they were treated normally each 4 hours for check-ups where the nurses will check the temperature and the pulses from the patient.

Blood sample need to be taken from the patient to confirm the diagnosis of the disease and identify the infectious parasites. For the blood sample to be taken, the ideal condition is when the body temperature of the patient is at the highest level. Experienced doctor will know the certain ideal time when to take the blood from the patient. However, the case is not the same with inexperienced doctors. What makes it worrying is that if they miss to take the blood when the temperature is at the highest level. What makes it even worst is that by missing the ideal time to draw the blood which will slower the diagnosis process, it will be vital to the patient's life.

This is where the "Body temperature detection system for malaria fever patient using RFID and GSM network" comes to the picture. The patient will be receiving the quickest treatment as the system will send signals anytime the temperature exceeds the limits. The signals will then notify the nurses through alarm at nurse's station and send via SMS to the patient's doctor.

The system is new, and therefore this research is made to survey the feasibility of the project for future uses. The questionnaire below is divided into 3 sections which are A, B and C. Please answer the questions by referring to every section's instructions. To simplify and save time, point forms are encouraged.

Section A: General / Background Information

Respondent can tick more than one for each [ ] provided or fill in the blanks.

I. Respondent Information:

1. How old are you?
   [ ] 10-20 years [ ] 20-30 years [ ] 30-40 years [ ] 50 years and above

II. Malaria Fever Knowledge and Experience Information:

1. Do you know about malaria fever and the conditions of the Malaria fever patient?
   [ ] Yes [ ] No [ ] Not Sure

2. Do you know that malaria fever can cause fatality?
   [ ] Yes [ ] No [ ] Not Sure

3. Have you ever catch malaria fever?
   [ ] Yes [ ] No

4. Have you experience people around you that suffering from malaria fever?
   [ ] Yes [ ] No

Section B: Criteria for selecting the best intensive care for malaria fever patient.

Respondent can tick more than one for each [ ] provided or fill in the blanks.

1. If you just catch malaria fever, where would you like to be treated?
   [ ] Hospitals
   [ ] Clinics
   [ ] Home
   [ ] Not sure

2. Do you know any method of intensive care for dengue fever patient other than manual check-ups?
   [ ] Yes [ ] No [ ] Not Sure

   If you answer **Yes,** please state the name of the system(s).

   i) _____
   ii) _____

3. Does every 4hours for manual check-ups are sufficient enough?
   [ ] Yes [ ] No [ ] Not Sure

4. Will it be safe to leave the patient in the ward without automatic updates of his/her body temperature?
   [ ] Yes [ ] No [ ] Not Sure

5. Can you trust that the nurses and doctors will arrive in time for treatment of patient when the body temperature arose?
   [ ] Yes [ ] No [ ] Not Sure

Section C: Feasibility study

Respondent can tick more than one for each [ ] provided or fill in the blanks.

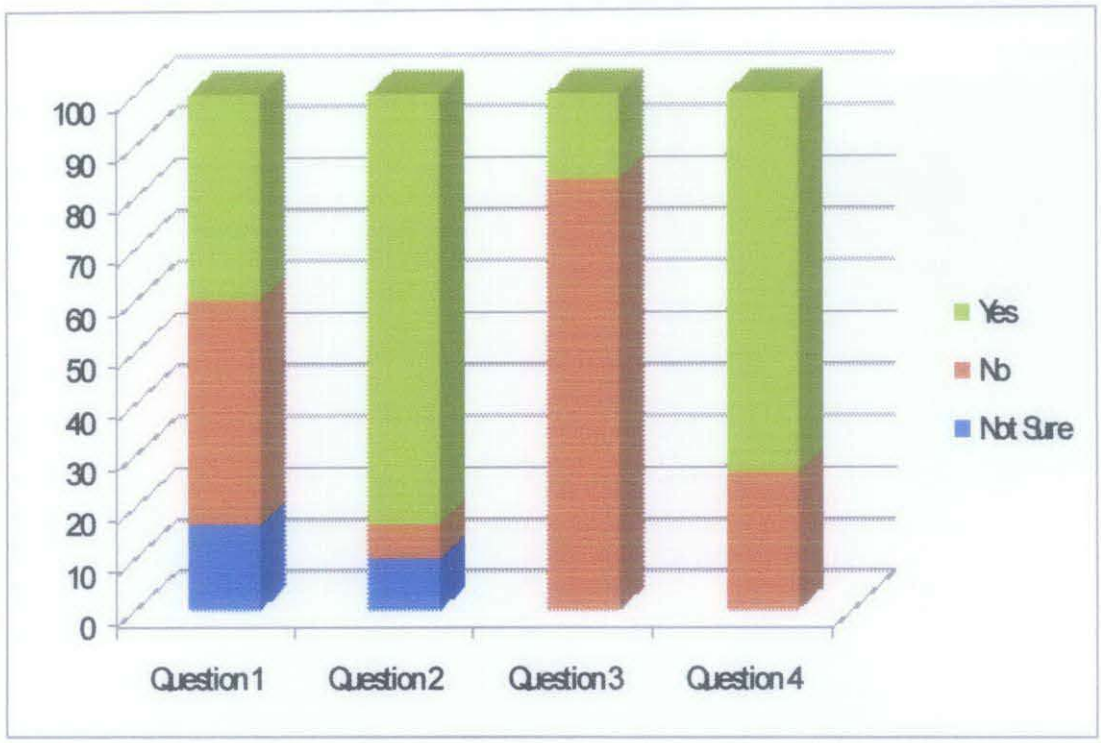
1. Will you pay more for your safety if you catch malaria fever?
   [ ] Yes [ ] No [ ] Not Sure

2. Will you go to hospitals that offer "Body temperature detection system for malaria fever patient using RFID and GSM network"?
   [ ] Yes [ ] No [ ] Not Sure

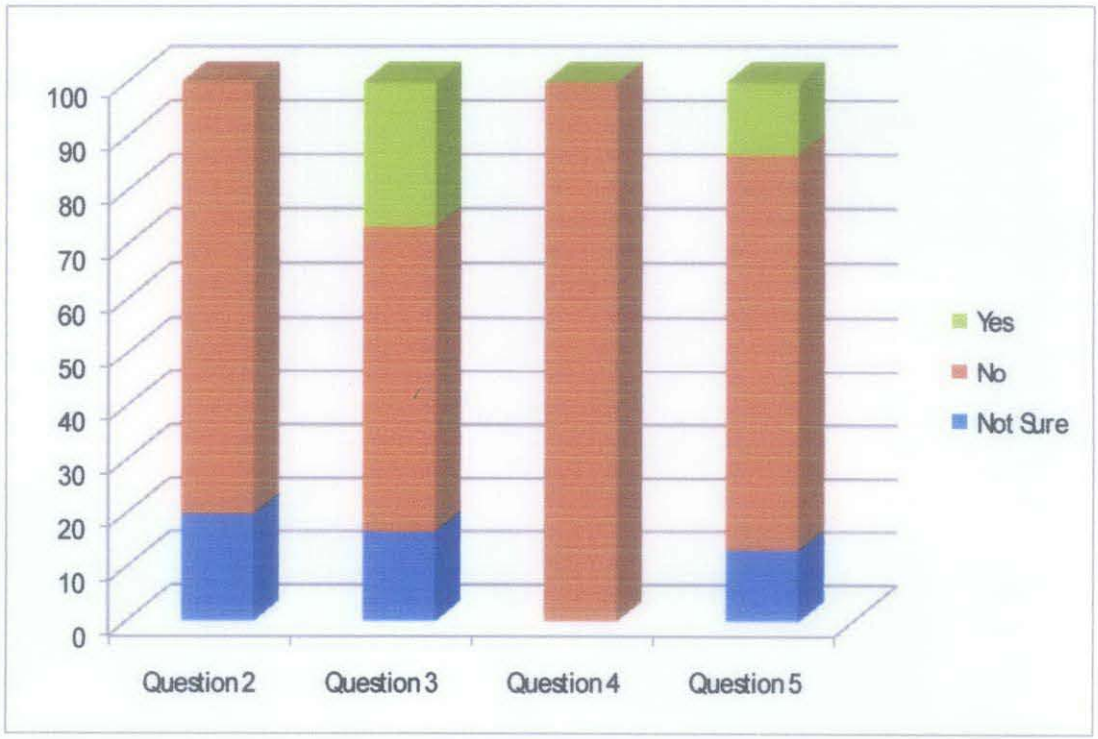
Section D: Suggestions

Do you have any suggestions on how to improve this questionnaire? Please feel free to write it on the space below.

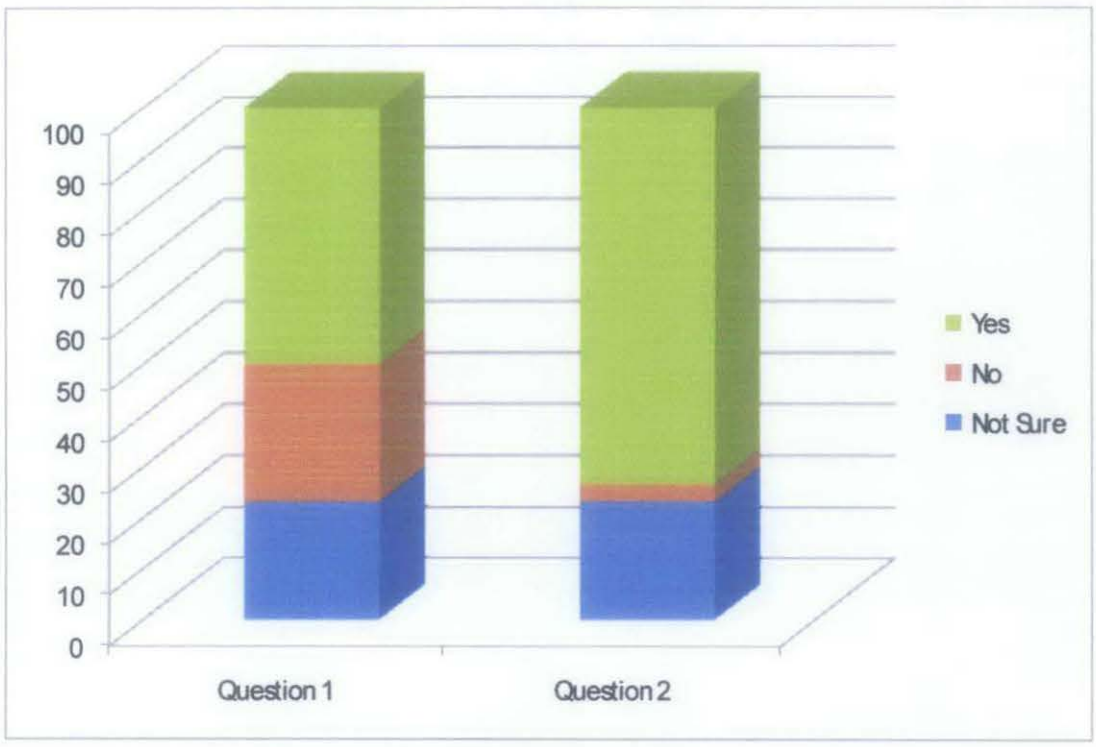
Thank you for your time and cooperation in completing the questionnaire. Your response will be used for research purpose only. It would be appreciated if you could return this questionnaire as soon as possible.



Result for Section A (II)



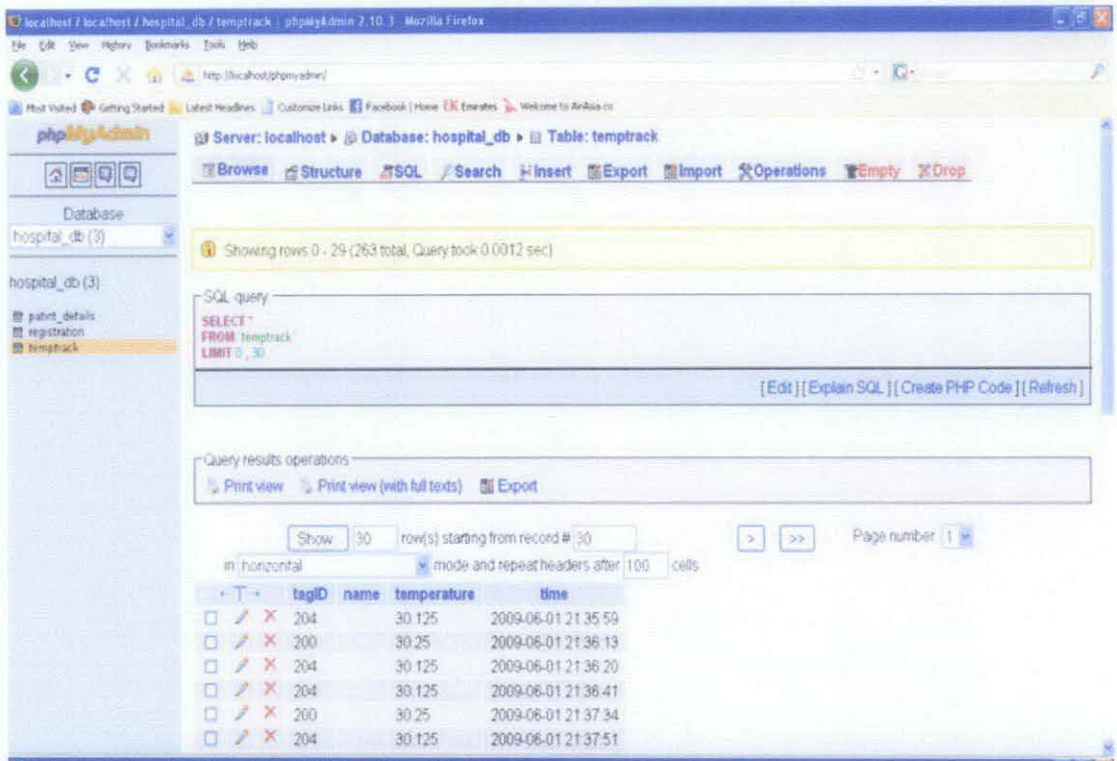
Result for Section B

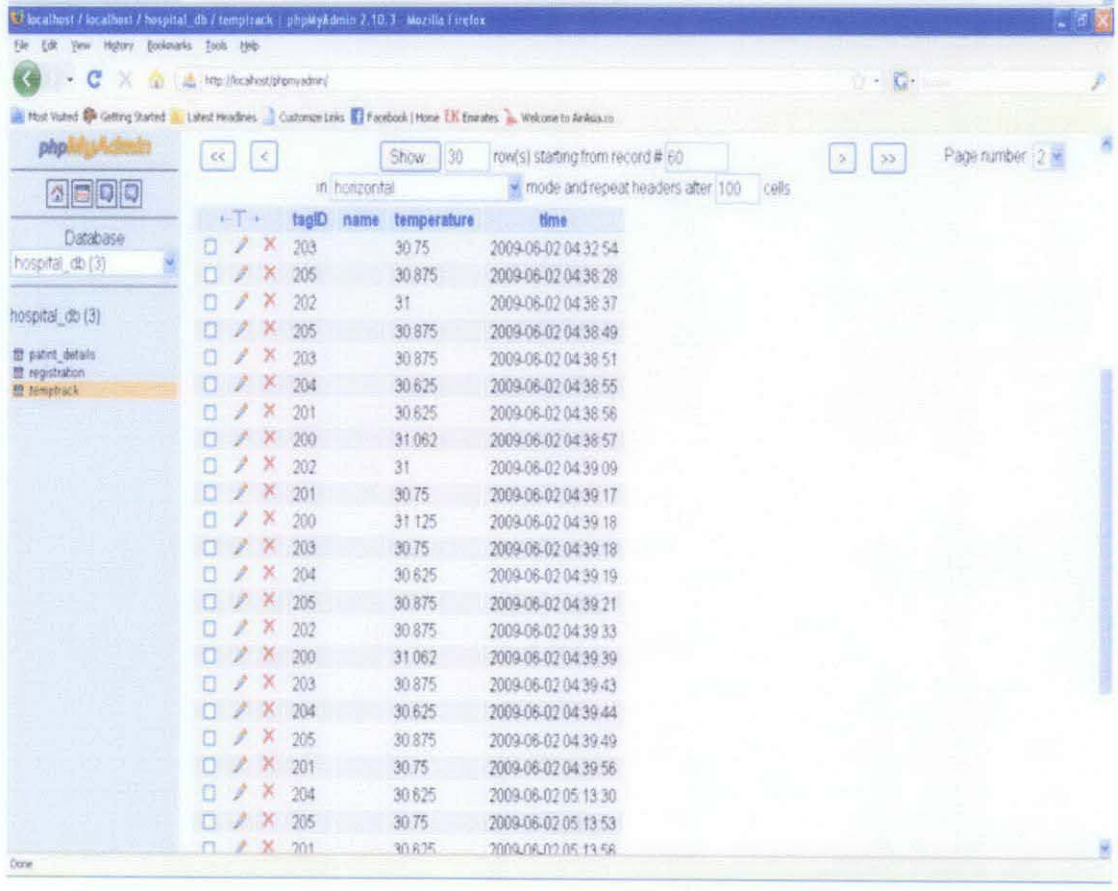


Result for Section C

APPENDIX C

DATABASE TABLES





APPENDIX D

VISUAL BASIC SOURCE CODE

i) Login Page Code

```
Imports System.IO
Imports MySql.Data.MySqlClient

---

Public Class login

  Private Sub login_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

  End Sub

---

  Private Function checkPswd(ByVal name As String, ByVal pswd As String) As Boolean

    Dim found As Boolean = False
    Dim rows As Integer = 0
    Dim myConnString As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    Dim cn As New MySqlConnection(myConnString)
    Dim cmdGH As New MySqlCommand("SELECT * FROM registration WHERE
user_name = '" & name & "'AND password = '" & pswd & "'", cn)
    Dim daGH As New MySqlDataAdapter
    Dim dsGH As New DataSet
    Dim dtGH As New DataTable
    cn.Open()
    Try
      daGH.SelectCommand = cmdGH
      daGH.Fill(dsGH, "employees")
      dtGH = dsGH.Tables("employees")
      rows = dtGH.Rows.Count()
    Catch ex As Exception
      MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
    End Try
    cn.Close()
    If (rows > 0) Then
      found = True
    Else
      found = False
    End If
    Return found
  End Function

---
```

```
  Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLogin.Click
    If (checkPswd(txtUname.Text, txtPswd.Text) = True) Then
      Main_Menu.Show()
      Me.Hide()
    Else
      MessageBox.Show("Incorrect User name or Password")
    End If
  End Sub
```

```
  Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
    txtUname.Text = ""
    txtPswd.Text = ""
  End Sub
```

```
  Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    Me.Close()
  End Sub
End Class
```

---

ii) Main Menu

```
Public Class Main_Menu
  Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

    Dim drExit As DialogResult
    drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Close()
    End If
  End Sub
```

```
  Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Go to patient Records ?", "Patient Records", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      Patient_Records.Show()
    End If
```

```
  End Sub
-----------------------------------------------------------------------
  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("go to Aministartion module ?", "Administration", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      System_admin.Show()
    End If
  End Sub
-----------------------------------------------------------------------
  Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
  End Sub
-----------------------------------------------------------------------
  Private Sub Main_Menu_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
  End Sub
-----------------------------------------------------------------------
  Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConnectStart.Click
    ConnStart.Show()
  End Sub
-----------------------------------------------------------------------
  Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label1.Click
  End Sub
End Class
-----------------------------------------------------------------------
```

iii) Administration Menu

```
Public Class System_admin
  Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Close()
    End If
  End Sub
-----------------------------------------------------------------------
```

```
  Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Back to Main Menu ?", "Back Conformation", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      Main_Menu.Show()
    End If
  End Sub
```

```
  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Add New User ?", "Add User", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      add_admin.Show()
    End If
  End Sub
```

```
  Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Add New Patient", "Add Patient", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      Patient_reg.Show()
    End If
  End Sub
  End Class
```

---

iv) Add Admin

```
Imports MySql.Data.MySqlClient
```

```
Public Class add_admin
  Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
```

```
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Close()
    End If
  End Sub

---

  Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Back to Main Menu ?", "Back Conformation", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      Main_Menu.Show()
    End If
  End Sub

---

  Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    txtIDnum.Clear()
    txtName.Clear()
    'txtPswd1.Clear()
    txtpswd.Clear()
    txtUname.Clear()
    txtIDnum.Focus()
  End Sub

---

  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim name As String = txtName.Text.ToString
    Dim rowsCheck As Integer = 0
    Dim ID As String = txtIDnum.Text.ToString
    Dim Post As String = cboPost.Text.ToString
    Dim user_name As String = txtUname.Text.ToString
    Dim Password As String = txtpswd.Text.ToString
    Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    'Connect to database
    Dim cn As New MySqlConnection(strCn)
    'Open database
    Dim cmdCheck As New MySqlCommand("SELECT * FROM registration WHERE
user_name = '" & user_name & "'", cn)
    Dim daCheck As New MySqlDataAdapter
    Dim dsCheck As New DataSet
    Dim dtCheck As New DataTable
    cn.Open()
```

```
        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()
            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try
        'If already connected, close connection
        If CBool(ConnectionState.Open) Then
            cn.Close()
        End If
        If rowsCheck = 0 Then
            'Open connection
            cn.Open()
            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO registration (ID, Post,
user_name, Password, name) VALUES ('" & ID & "','" & Post & "','" & user_name & "','" &
Password & "','" & name & "')", cn)
            cmdIns.ExecuteNonQuery()
            If CBool(ConnectionState.Open) Then
                cn.Close()
            End If
            'If registration successful, show ID and name
            MsgBox(name + "/" + ID + " has been registered successfully")
        Else
            'If already existing officer, show error
            MsgBox("Error!!! " + name + "is already registered in the database")
        End If
    End Sub
        End Class
```

---

v) Add Patient

```
Imports System.Data
Imports MySql.Data.MySqlClient
```

```
Public Class Patient_reg
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Dim drExit As DialogResult
        drExit = MessageBox.Show("Do you really want to Exit ?", "Exit Conformation", _
```

```
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Close()
    End If
  End Sub
```

```
  Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim drExit As DialogResult
    drExit = MessageBox.Show("Back to Main Menu ?", "Back Conformation", _
                MessageBoxButtons.YesNo, MessageBoxIcon.Warning,
MessageBoxDefaultButton.Button2)
    If drExit = DialogResult.Yes Then
      Me.Hide()
      Main_Menu.Show()
    End If
  End Sub
```

```
  Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    txtAge.Clear()
    txtdateofadmit.Clear()
    txtDiag.Clear()
    txtDob.Clear()
    txtICnum.Clear()
    txtName.Clear()
    txtPatID.Clear()
    txtTemp.Clear()
    cboTagID.ResetText()
    txtAge.Focus()
  End Sub
```

```
  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim name As String = txtName.Text.ToString
    Dim rowsCheck As Integer = 0
    Dim icnum As String = txtICnum.Text.ToString
    Dim diag As String = txtDiag.Text.ToString
    Dim dob As String = txtDob.Text.ToString
    Dim temp As String = txtTemp.Text.ToString
    Dim tagid As String = cboTagID.Text.ToString
    Dim patid As String = txtPatID.Text.ToString
    Dim doa As DateTime = DateTime.Now
    Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    'Connect to database
    Dim cn As New MySqlConnection(strCn)
```

```
        'Open database
        Dim cmdCheck As New MySqlCommand("SELECT * FROM patint_details WHERE
tagid_patid = '" & patid & "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable
        cn.Open()
        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()
            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try
        'If already connected, close connection
        If CBool(ConnectionState.Open) Then
            cn.Close()
        End If
        If rowsCheck = 0 Then
            'Open connection
            cn.Open()
            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO patint_details (name,
ic_num, tagid_patid, dateofbirth, dateofadmin, diagnosis, temperature) VALUES ('" & name
& "','" & icnum & "','" & tagid & "','" & dob & "','" & Format(DateTime.Now, "yyyy-MM-dd
HH:mm:ss") & "','" & diag & "','" & temp & "')", cn)
            cmdIns.ExecuteNonQuery()
            If CBool(ConnectionState.Open) Then
                cn.Close()
            End If
            'If registration successful, show ID and name
            MsgBox(name + "/" + icnum + " has been registered successfully")
        Else
            'If already existing officer, show error
            MsgBox("Error!!! " + name + "is already registered in the database")
        End If
    End Sub
End Class
```

---

vi) Start Option

```
Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient
Imports System.Threading
```

```
Public Class ConnStart

  Dim Hconn As IntPtr
  Dim readerIP(20) As Byte
  Dim readerPort As UInt16
  Dim commPort As UInt32
  Dim commBaud As UInt32
  Dim myPKTID As Integer
  Dim registered As Boolean
  Dim ActiveWaveAPI As AW_API_NET.APINetClass = New
AW_API_NET.APINetClass
  Dim ReaderEventHandler As AW_API_NET.fReaderEvent
  Dim TagEventHandler As AW_API_NET.fTagEvent
  Dim pubReaderID As Integer
  Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
```

```
  Private Sub ConnStart_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    readerPort = Convert.ToUInt16(10001)
    commPort = Convert.ToUInt32(1)
    commBaud = Convert.ToUInt32(115200)
    ' Initialize callback functions
    ReaderEventHandler = New AW_API_NET.fReaderEvent(AddressOf
Me.OnReaderEvent)  'AddressOf Me.OnReaderEvent
    TagEventHandler = New AW_API_NET.fTagEvent(AddressOf Me.OnTagEvent)
    registered = False
  End Sub
```

```
  Private Function OnReaderEvent(ByVal readerEvent As AW_API_NET.rfReaderEvent_t)
As Integer
    Dim ipStr As String = ""
    'Beep()
    If
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SCAN_N
ETWORK)) Then
      For i As Integer = 0 To readerEvent.ip.Length - 1
```

```
            ipStr += Convert.ToChar(readerEvent.ip(i))
        Next i
        IPListBox.Items.Add(ipStr)
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_OPEN_SO
CKET)) Then
        For i As Integer = 0 To readerEvent.ip.Length - 1
            ipStr += Convert.ToChar(readerEvent.ip(i))
        Next i
        'AddMsg("Socket Opened IP = " + ipStr)
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_CLOSE_S
OCKET)) Then
        For i As Integer = 0 To readerEvent.ip.Length - 1
            ipStr += Convert.ToChar(readerEvent.ip(i))
        Next i
        'AddMsg("Socket Closed IP = " + ipStr)
        Dim index As Integer
        If ipStr.Length > 0 Then
            index = IPListBox.FindStringExact(ipStr)
            If index >= 0 Then
                IPListBox.Items.RemoveAt(index)
            End If
        End If
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_STD_FGE
N_POWERUP)) Then
        'FGenIDTextBox.Text = readerEvent.fGenerator.ToString()
        'AddMsg("STD FGen Powered UP")
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_READER
_POWERUP)) Then
        'ReaderIDTextBox.Text = readerEvent.reader.ToString()
        'tText(readerEvent.reader.ToString())
        'AddMsg("Reader Powered UP")
    ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_QUERY_
STD_FGEN)) Then
        Dim str As String
        'AddMsg(readerEvent.eventType.ToString)
        'AddMsg(AW_API_NET.APIConsts.RF_READER_POWERUP.ToString)
        'AddMsg("STD FGEN Query _____")
        str = readerEvent.smartFgen.fsValue
        'AddMsg("FS Value = " + str)
        str = readerEvent.smartFgen.txTime
        'AddMsg("TX Time = " + str)
        str = readerEvent.smartFgen.waitTime
        'AddMsg("Wait Time = " + str)
```

```
        str = readerEvent.smartFgen.assignRdr
        'AddMsg("Assigned Rdr = " + str)
      ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_GET_RD
R_FS)) Then
        'FSTextBox.Text = readerEvent.smartFgen.fsValue.ToString()
        'AddMsg("Reader ID:" + readerEvent.reader.ToString() + "  FS:" +
readerEvent.smartFgen.fsValue.ToString())
      ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SET_RDR
_FS)) Then
        'AddMsg("Reader FS was set successfuly")
      ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SCAN_IP)
) Then
        ipStr = GetStringIP(readerEvent.ip)
        If ipStr.Length > 0 Then
          If IPListBox.FindStringExact(ipStr) = -1 Then
            IPListBox.Items.Add(ipStr)
          End If
          'AddMsg("ScanIP IP=" + ipStr)
        End If
      End If
      ReportReaderEvent(readerEvent)
      Return 0
    End Function

---

    Private Sub ReportReaderEvent(ByRef readerEvent As AW_API_NET.rfReaderEvent_t)
      If readerEvent.errorStatus.ToString() = 0 Then
        commStatus.Text = "ON"
        pubReaderID = readerEvent.reader.ToString()
      End If
    End Sub

---

    Private Function OnTagEvent(ByVal tagEvent As AW_API_NET.rfTagEvent_t) As
Integer
      Dim ipStr As String = ""
      If
tagEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_TAG_READ)
) Then
        Dim str As String
        Dim n As Integer
        n = CInt(Convert.ToInt16(tagEvent.tag.dataLen))
        For i As Integer = 0 To n - 1
          str = tagEvent.tag.data(i)
          ipStr += str + ""
        Next
      End If
```

```
    ReportTagEvent(tagEvent)
    Return 0
  End Function
```

```
  Private Sub ReportTagEvent(ByRef tagEvent As AW_API_NET.rfTagEvent_t)
    If tagEvent.tag.id <> 0 Then
      If tagEvent.tag.temp.temperature <> 0 Then
        'MessageBox.Show(tagEvent.tag.id.ToString + "    " +
tagEvent.tag.temp.temperature.ToString)
        insertrow(tagEvent.tag.id.ToString(), tagEvent.tag.temp.temperature.ToString,
tagEvent.errorStatus.ToString())
      End If
    End If
  End Sub
```

```
  Private Sub insertrow(ByVal tgid As Integer, ByVal temp As String, ByVal st As Integer)
    Dim cn As New MySqlConnection(strCn)
    Dim prvTime As DateTime = Nothing
    Dim name = Nothing
    If st = 0 Then
      Dim prvTimeCmd As New MySqlCommand("SELECT time FROM temptrack
WHERE tagID = '" & tgid.ToString & "'", cn)
      Dim prvTimeRdr As MySqlDataReader
      cn.Open()
      prvTimeRdr = prvTimeCmd.ExecuteReader
      While prvTimeRdr.Read
        prvTime = prvTimeRdr.Item("time")
      End While
      cn.Close()
      If (Format(prvTime, "yyyy-MM-dd HH:mm:ss") <
Format(DateTime.Now.AddSeconds(-20), "yyyy-MM-dd HH:mm:ss")) Then
        Dim insCmd As New MySqlCommand("INSERT INTO temptrack (tagID, name,
temperature, time) VALUES ('" & tgid.ToString & "', '" & name & "', '" & temp & "', '" &
Format(DateTime.Now, "yyyy-MM-dd HH:mm:ss") & "')", cn)
        cn.Open()
        insCmd.ExecuteNonQuery()
        cn.Close()
        MessageBox.Show(tgid.ToString + "    " + temp)
      End If
    End If
  End Sub
```

```
  Public Function GetStringIP(ByVal ip As Byte()) As String
    Dim p As Integer
    Dim s As String
    Dim ct As Integer
    ct = 0
    p = 0
```

```
    s = ""
    While (Convert.ToBoolean((ct <= 3)) AndAlso Convert.ToBoolean((p < 20)) AndAlso
Convert.ToBoolean((ip(p) <> 0)))
      If ip(p) <> 46 Then
        s += Convert.ToString(ip(p) - 48) '- 48
        p += 1
      Else
        ct += 1
        p += 1
        s += "."
      End If
    End While
    Return s
  End Function
```

```
  Private Sub readConfig()
    Try
      ' Create an instance of StreamReader to read from a file.
      Using sr As StreamReader = New StreamReader("config.txt")
        Dim line As String
        ' Read and display the lines from the file until the end
        ' of the file is reached.
        frmConfig.lbIp.Items.Clear()
        line = sr.ReadLine()
        If Mid(line, 1, 3) = "ip:" Then
          frmConfig.lbIp.Items.Add(Mid(line, 4))
        End If
        'lbIp.Items.Clear()
        While Not line Is Nothing
          line = sr.ReadLine()
          If Mid(line, 1, 3) = "ip:" Then
            frmConfig.lbIp.Items.Add(Mid(line, 4))
          End If
        End While
        sr.Close()
      End Using
    Catch E As Exception
      ' Let the user know what went wrong.
      Console.WriteLine("The file could not be read:")
      Console.WriteLine(E.Message)
    End Try
  End Sub
```

```
  Private Sub BtnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConnect.Click
    scanIP()
    commStatus.BackColor = Color.DarkSeaGreen
  End Sub
```

```
Private Sub scanIP()
    Dim iRet As Integer
    Dim ip(20) As Byte
    Dim stripx As String
    readIp()
    If (myPKTID >= 223) Then
      myPKTID = 1
    Else
      myPKTID = myPKTID + 1
    End If
    If registered = False Then
      ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
      ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
      registered = True
    End If
    For c As Integer = 0 To frmConfig.lbIp.Items.Count - 1
      stripx = frmConfig.lbIp.Items.Item(c).ToString
      For i As Integer = 0 To stripx.ToString.Length - 1
        ip(i) = Convert.ToByte(stripx.ToString.Chars(i))
      Next i
      iRet = ActiveWaveAPI.rfScanIP(ip, Convert.ToUInt16(myPKTID))
    Next c
    OpenConn()
  End Sub
```

```
  Private Sub OpenConn()
    Dim iRet As Integer
    Dim ip(20) As Byte
    Dim cIP(20) As Char
    Dim stripx As String = "192.168.10.26"
    If (myPKTID >= 223) Then
      myPKTID = 1
    Else
      myPKTID = myPKTID + 1
    End If
    iRet = ActiveWaveAPI.rfOpenSocket(readerIP, readerPort, False,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS), Convert.ToUInt16(myPKTID))
    If (myPKTID >= 223) Then
      myPKTID = 1
    Else
      myPKTID = myPKTID + 1
    End If
    iRet = ActiveWaveAPI.rfResetReader(UInt16.Parse(1), UInt16.Parse(0),
UInt16.Parse(0), Convert.ToUInt16(AW_API_NET.APIConsts.ALL_READERS),
Convert.ToUInt16(myPKTID))
  End Sub
```

```
Private Sub readIp()
    Try
        Using sr As StreamReader = New StreamReader("config.txt")
            Dim line As String
            line = sr.ReadLine()
            If Mid(line, 1, 3) = "ip:" Then
                frmConfig.lbIp.Items.Add(Mid(line, 4))
            End If
            While Not line Is Nothing
                line = sr.ReadLine()
                If Mid(line, 1, 3) = "ip:" Then
                    frmConfig.lbIp.Items.Add(Mid(line, 4))
                End If
            End While
            sr.Close()
        End Using
    Catch E As Exception
        ' Let the user know what went wrong.
        Console.WriteLine("The file could not be read:")
        Console.WriteLine(E.Message)
    End Try
End Sub
```

```
Private Sub CloseConn()
    Dim iRet As Integer
    Dim ip(20) As Byte
    Dim cIP(20) As Char
    If (myPKTID >= 223) Then
        myPKTID = 1
    Else
        myPKTID = myPKTID + 1
    End If
    'If AllIPRadioButton.Checked Then
    iRet = ActiveWaveAPI.rfCloseSocket(readerIP,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS))
    IPListBox.Items.Clear()
End Sub
```

```
Private Sub btnConfig_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConfig.Click
    frmConfig.Show()
End Sub
```

```
Private Sub tmrTemp_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrTemp.Tick
    rqstTemp()
End Sub
```

```
  Private Sub rqstTemp()
    Dim iRet As Integer
    Dim tagSelect As AW_API_NET.rfTagSelect_t
    Dim tagList(50) As UInt32
    Dim rdrID As UInt16
    Dim longInterval As Boolean
    Dim RdrCmdType As Integer
    rdrID = UInt16.Parse(0)
    RdrCmdType = APIConsts.ALL_READERS
    tagSelect.tagList = tagList
    tagSelect.tagList(0) = UInt32.Parse(0)
    tagSelect.numTags = Convert.ToUInt32(50)
    tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)
    tagSelect.tagType = APIConsts.ACCESS_TAG
    If (myPKTID >= 223) Then
      myPKTID = 1
    Else
      myPKTID = myPKTID + 1
    End If
    longInterval = False
    'iRet = ActiveWaveAPI.rfCallTags(UInt16.Parse(1), rdrID, UInt16.Parse(0),
UInt16.Parse(0), tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))
    'iRet = ActiveWaveAPI.rfQueryTags(UInt16.Parse(1), rdrID, UInt16.Parse(0),
tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))
    iRet = ActiveWaveAPI.rfGetTagTemp(UInt16.Parse(1), rdrID, UInt16.Parse(0),
tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))
  End Sub
```

```
  Private Sub BtnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnStart.Click
    tmrTemp.Enabled = True
  End Sub
```

```
  Private Sub BtnHide_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnHide.Click
    Me.Hide()
  End Sub
      End Class
```

---

vii) Patient Records

```
Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient
Imports System.Threading

---

Public Class Patient_Records
  Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
  Dim temper As ListViewItem
  Dim tgID As String
  Private str2 As System.Threading.Thread
  Private str1 As System.Threading.Thread

---

  Private Sub Patient_Records_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = False
    lvwTemp.Columns.Clear()
    lvwTemp.Columns.Add("Temperature", 100, HorizontalAlignment.Left)
    lvwTemp.Columns.Add("Date                    Time", 225, HorizontalAlignment.Right)
    lvwTemp.Hide()
  End Sub

---

  Private Sub listDetails()
    Dim name As String = ""
    Dim diagn As String = ""
    Dim birthD As DateTime = Nothing
    Dim age As Integer = 0
    Dim admisD As DateTime = Nothing
    Dim ic As String = ""
    tgID = txtID.Text
    If (checkID(tgID) = True) Then
      If Me.InvokeRequired Then
        Me.Invoke(New MethodInvoker(AddressOf listDetails))
      Else
        'Establish connection
        Dim temp As String = ""
        Dim dat As DateTime = Nothing
        Dim cn As New MySqlConnection(strCn)
        'Fill in Patient details
        Dim cmdDetails As New MySqlCommand("SELECT * FROM patint_details
WHERE tagid_patid = '" & tgID & "'", cn)
        Dim rdrDetails As MySqlDataReader
        cn.Open()
```

```
        rdrDetails = cmdDetails.ExecuteReader
        While rdrDetails.Read
          name = rdrDetails.Item("name")
          ic = rdrDetails.Item("ic_num")
          birthD = rdrDetails.Item("dateofbirth")
          diagn = rdrDetails.Item("diagnosis")
          age = rdrDetails.Item("age")
          admisD = rdrDetails.Item("dateofadmin")
        End While
        rdrDetails.Close()
        cn.Close()
        txtname.Text = name
        txtage.Text = age.ToString
        txtdiag.Text = diagn
        txtdoa.Text = Format(admisD, "yyyy-MM-dd HH:mm:ss")
        txtdob.Text = Format(birthD, "yyyy-MM-dd HH:mm:ss")
        txticnum.Text = ic
        txtname.Show()
        txtage.Show()
        txtdoa.Show()
        txtdiag.Show()
        txticnum.Show()
        txtdob.Show()
        Label3.Show()
        Label4.Show()
        Label5.Show()
        Label6.Show()
        Label7.Show()
        Label8.Show()
        btnConfirm.Show()
      End If
    Else
      MessageBox.Show("There is no record corresponding to this ID")
      txtname.Text = ""
      txtage.Text = ""
      txtdoa.Text = ""
      txtdiag.Text = ""
      txticnum.Text = ""
      txtdob.Text = ""
      lvwTemp.Items.Clear()
      lvwTemp.Hide()
      btnConfirm.Hide()
    End If
  End Sub

---

  Sub Thread1()
    listDetails()
  End Sub
```

```
  Private Function checkID(ByVal id As String) As Boolean
    Dim found As Boolean = False
    Dim rows As Integer = 0
    Dim myConnString As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
    Dim cn As New MySqlConnection(myConnString)
    Dim cmdGH As New MySqlCommand("SELECT * FROM patint_details WHERE
tagid_patid = '" & id & "'", cn)
    Dim daGH As New MySqlDataAdapter
    Dim dsGH As New DataSet
    Dim dtGH As New DataTable
    cn.Open()
    Try
      daGH.SelectCommand = cmdGH
      daGH.Fill(dsGH, "details")
      dtGH = dsGH.Tables("details")
      rows = dtGH.Rows.Count()
    Catch ex As Exception
      MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
    End Try
    If ConnectionState.Open Then
      cn.Close()
    End If
    If (rows > 0) Then
      found = True
    Else
      found = False
    End If
    Return found
  End Function
```

```
  Private Sub listvw()
    Dim i As Integer = 0
    tgID = txtID.Text
    Dim cn As New MySqlConnection(strCn)
    If Me.InvokeRequired Then
      Me.Invoke(New MethodInvoker(AddressOf listvw))
    Else
      'Populate the ListView
      Dim cmd As New MySqlCommand("SELECT * FROM temptrack WHERE tagID =
'" & tgID & "'", cn)
      Dim rdrTemp As MySqlDataReader
      lvwTemp.Items.Clear()
      If cn.State = ConnectionState.Closed Then
        cn.Open()
      End If
      'Display raw in listview (attendlist)
```

```
        rdrTemp = cmd.ExecuteReader()
        While rdrTemp.Read
          temper = lvwTemp.Items.Add(rdrTemp.Item("temperature"))
          temper.SubItems.Add(Format(rdrTemp.Item("time"), "yyyy-MM-dd
HH:mm:ss"))
        End While
        rdrTemp.Close()
        cn.Close()
        'coloring backgroung
        While i <= lvwTemp.Items.Count - 1
          If i Mod 2 = 0 Then
            lvwTemp.Items(i).BackColor = Color.Aquamarine
          Else
            lvwTemp.Items(i).BackColor = Color.White
          End If
          i = i + 1
        End While
        lvwTemp.Show()
      End If
    End Sub
```

```
    Sub Thread2()
      listvw()
    End Sub
```

```
    Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSearch.Click
      str1 = New Thread(AddressOf Thread1)
      str1.Start()
    End Sub
```

```
    Private Sub btnConfirm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConfirm.Click
      str2 = New Thread(AddressOf Thread2)
      str2.Start()
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
      Main_Menu.Show()
      Me.Hide()
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
      Me.Close()
    End Sub
          End Class
```

viii) GSM Coding

```
Private Function DBConnect() As String()
  Dim Conn As New ADODB.Connection
  Dim sqlString As String
  Dim RS As ADODB.Recordset
  Dim DS As String
  Dim currdir As String
  Dim Details(3) As String 'details are in array
  Dim DID As Integer
  Directory = "C:\Documents and Settings\user\Desktop\GSM Coding\Cubaan\"
  DS = Directory & "db.mdb"
  Conn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source =" &
DS
  Conn.Open
  sqlString = "SELECT * FROM Patient" 'taken from database at 'Patient' table
  Set RS = Conn.Execute(sqlString)
  Details(0) = RS.Fields("PatientName").Value
  DID = RS.Fields("DoctorID").Value
  Conn.Close
  Conn.Open
  sqlString = "SELECT * FROM Doctor WHERE DoctorID=" & DID 'taken from
database at 'Doctor' table
    Set RS = Conn.Execute(sqlString)
  Details(1) = RS.Fields("DoctorName").Value
  Details(2) = RS.Fields("DoctorHP").Value
    Conn.Close
  DBConnect = Details()
End Function
```

```
Private Sub CmdSend_Click()
  Dim Details() As String
  Details = DBConnect()
  Message = "The Patient: " & Details(0) & " Temperature is Critical!"
  Alert.Text = Message
  MSComm1.CommPort = 12   'switch on the port, change accordingly with your
PC
  MSComm1.PortOpen = True
  Shape1.FillColor = vbBlue
  MSComm1.Settings = "9600,n,8,1"
  MSComm1.Output = "AT+CMGF=1" & vbCrLf
  MSComm1.Output = "AT+CMGS=" & Chr(34) & Details(2) & Chr(34) & vbCrLf
  MSComm1.Output = Message & Chr(26)
  MSComm1.PortOpen = False
End Sub
```
